This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PR-MaGIC: Prompt Refinement Via Mask Decoder Gradient Flow For In-Context Segmentation

Minjae Lee*, Sungwoo Hur*, Soojin Hwang, Won Hwa Kim
Pohang University of Science and Technology, Pohang, South Korea
{lalswo010,hursungwoo,soojin0622,wonhwa}@postech.ac.kr
https://postech-minjaelee.github.io/PR-MaGIC/

Abstract

*Visual Foundation Models (VFMs) such as the Segment Anything Model (SAM) have significantly advanced broad use of image segmentation. However, SAM and its variants necessitate substantial manual effort for prompt generation and additional training for specific applications. Recent approaches address these limitations by integrating SAM into in-context (one/few shot) segmentation, enabling auto-prompting through semantic alignment between query and support images. Despite these efforts, they still generate sub-optimal prompts that degrade segmentation quality due to visual inconsistencies between support and query images. To tackle this limitation, we introduce PR-MaGIC (**P**rompt **R**efinement via **Ma**sk Decoder **G**radient Flow for **I**n-**C**ontext Segmentation), a training-free test-time framework that refines prompts via gradient flow derived from SAM's mask decoder. PR-MaGIC seamlessly integrates into in-context segmentation frameworks, being theoretically grounded yet practically stabilized through a simple top-1 selection strategy that ensures robust performance across samples. Extensive evaluations demonstrate that PR-MaGIC consistently improves segmentation quality across various benchmarks, effectively mitigating inadequate prompts without requiring additional training or architectural modifications.*

1. Introduction

Large models have paved a new era of foundation models, e.g., Large Language Models (LLMs) [2, 27, 29], demonstrating exceptional versatility and generality across various tasks. In computer vision, Visual Foundation Models (VFMs) are transforming tasks such as image classification [7, 26], object detection [3, 15], and segmentation [6, 23] by leveraging large-scale data and advanced architectures. Specifically in segmentation, the Segment Anything Model (SAM) [11] addresses the challenges of segmenting objects with diverse appearance by taking prompts such as points, bounding boxes, and coarse masks.

*Equal contribution.

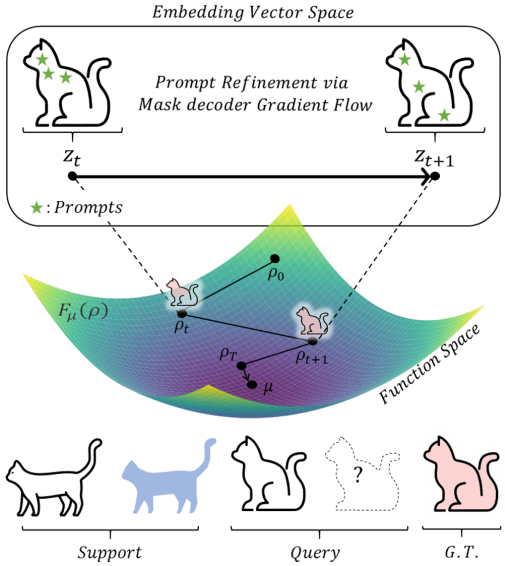
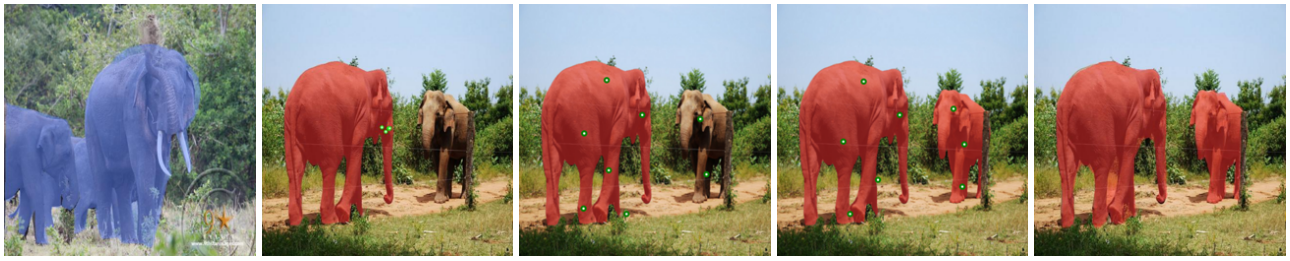


Figure 1. **Illustration of PR-MaGIC.** PR-MaGIC iteratively refines prompts for segmentation by updating the embedding vector distribution ρ_t with mask decoder gradient flow. This process minimizes the KL divergence between ρ_t and the target embedding vector distribution μ from the ground truth mask. At each iteration t, given an initial set of prompts and their corresponding segmentation masks, the embedding vector z_t is updated along the gradient flow derived from the mask decoder, which in turn updates ρ_t.

While SAM is highly versatile, it often produces imprecise boundaries and small holes under complex conditions, as it was trained for general object segmentation at relatively low resolution. Initial efforts fine-tuned SAM to improve segmentation precision across various scenarios [4, 10, 28]. However, these methods still substantially depend on additional parameters, extra curated data, and manual effort to obtain suitable prompts. Recent works integrate VFMs into one/few-shot (in-context) segmentation to enable auto-prompting via support–query semantic alignment. PerSAM [32] uses a single support image with a rough mask to build similarity-based prompts, and PerSAM-F [32] further fine-tunes SAM to linearly combine hierarchical masks to



(a) Support Image (b) PerSAM-F (c) Matcher (d) PR-MaGIC (e) Ground Truth

Figure 2. **Example of prompts (green dots) and segmentation results (red masks).** (a) Support Image (elephant), (b) and (c) Misaligned prompts and segmentation results from PerSAM-F and Matcher, (d) Result from Matcher refined by PR-MaGIC, (e) Ground Truth.

reduce ambiguity. Matcher [14] assembles existing VFMs with sophisticated prompt sampling to solve one/few-shot segmentation without extra training.

Although the methods mentioned above have been successful, variations in appearance (e.g., color, viewpoint, and shape) between query and support images often lead to misleading similarity maps. This causes auto-prompting segmentation frameworks (e.g., PerSAM-F [32] and Matcher [14]) to generate inadequate prompts, e.g., false-positive, semantically ambiguous, or semantically insufficient prompts, that confuse the SAM's decoder in capturing the target object, thereby compromising the quality of the segmentation mask.

This challenge motivates us to explore a refinement strategy that updates the initial suboptimal prompts produced by existing auto-prompting segmentation frameworks using mask decoder gradient flow. Such an approach leverages the rich information embedded in the decoder, which has been jointly trained with the encoder during SAM's large-scale pretraining. In this regime, we propose **P**rompt **R**efinement via **Ma**sk Decoder **G**radient Flow for **I**n-**C**ontext Segmentation (PR-MaGIC). PR-MaGIC is a *training-free*, inference-time framework that updates the query embedding via a mask–decoder–driven gradient flow, resamples prompts, and thereby improves the segmentation quality. The update is motivated by finding the distribution that yields the highest mask quality under the VFM's mask decoder, which aligns the query embedding with the target object's semantics during inference. Fig. 1 overviews the gradient flow, where the query embedding distribution is progressively refined.

Our theoretical analysis shows exponential convergence of the gradient flow *under a proximity assumption*: the initial embedding distribution lies near a decoder-optimal neighborhood. For the initial auto-prompts that do not comply with the assumption, we explore multiple refinement steps and adaptively select the most reliable mask to achieve stable performance across samples. The key contributions of this work are as follows:

- **Training-free test-time refinement via gradient flow.** We introduce a novel, training-free refinement method that updates the query embedding with a mask–decoder–driven gradient flow and resamples prompts, improving segmentation quality without learnable parameters, architectural changes, or extra data.
- **Segmentation mask selection for robustness.** We generate candidate masks across T iterations and propose a simple top-1 support–query similarity selection to choose the final mask, which safeguards against step-size sensitivity and sample-dependent instability.
- **Convergence analysis under a proximity assumption.** We provide a theoretical analysis showing that, *when the initial embedding distribution is near a decoder-optimal neighborhood*, the entropy-regularized KL gradient flow drives exponential convergence of the query embeddings. We further complement this with a sensitivity study indicating that the assumption can be fragile in practice, motivating our segmentation mask selection step.

We validate PR-MaGIC on six datasets and two tasks (semantic and part segmentation), demonstrating consistent gains as a plug-and-play module, particularly when the semantic gap between support and query is not excessive. A comparative exemplar result is given in Fig. 2, which illustrates that PR-MaGIC produces better prompts and more precise masks successfully capturing the complete target objects, unlike PerSAM-F and Matcher which tend to yield less effective prompts.

2. Related Works

2.1. One/Few-shot segmentation

Few-shot segmentation aims to develop models capable of segmenting new classes using only a few annotated samples [22], making it particularly useful in scenarios where acquiring large annotated data is difficult. Typically, these models use a pre-trained feature extractor to process both support and query images, combining information from support images to perform segmentation [9, 31].

Recently, several approaches leveraging SAM as a pre-trained model have been proposed. PerSAM relies solely on one-shot data, comprising a support image and a rough mask of the target object. It utilizes a similarity map between a query and target object embeddings to locate the target object precisely, facilitating prompt generation. How-

ever, PerSAM faces challenges when there is large visual ambiguity, such as objects with visually distinct subparts or hierarchical structures [32]. PerSAM-F, fine-tuning variant of PerSAM, addresses this problem by introducing learnable parameters to fine-tune SAM, producing a linear combination of potential segmentation masks of different hierarchical levels to resolve ambiguity [32]. This approach improves segmentation accuracy and prevents overfitting on one-shot data, demonstrating enhanced performance in challenging scenarios. Similarly, Matcher [14], effectively combines pre-existing VFMs to tackle one/few-shot segmentation tasks without additional training. Matcher utilizes DINOv2 [17] as an encoder to compute a similarity map with semantic understanding to accurately identify the location of a target object in the query image. It then employs SAM as a segmenter to obtain segmentation masks, benefiting from the zero-shot segmentation performance of the SAM.

2.2. Fine-tuning variants of SAM

For efficient adaptation of SAM to various downstream tasks, a range of fine-tuning variants has emerged. HQ-SAM [10], for instance, adds a high-quality output token and trains on the HQSeg-44K dataset to improve mask precision. VRP-SAM [25] implements a visual reference prompt encoder, requiring specific reference images and detailed training. SAM-Adapter [4] modifies the architecture by adding lightweight adapter layers while keeping most of SAM's original parameters frozen, reducing training effort. MobileSAM [30] replaces the heavy ViT-H encoder with Tiny-ViT, reducing the model size and complexity while still requiring adaptation.

All of these methods, however, necessitate additional training data and structural changes to the model architecture. These fine-tuning strategies still heavily depend on manual prompts, with training datasets requiring carefully crafted prompt annotations alongside segmentation data. This reliance on curated prompts limits their applicability in real-world settings, where obtaining accurate prompts and consistent user interaction may be challenging.

3. Method

This section presents PR-MaGIC, a training-free, test-time refinement framework for promptable segmentation. Starting from support and query embeddings, we run a *mask-decoder–driven gradient flow* that iteratively updates the query embedding and resamples prompts to produce a set of candidate masks; we then apply *Segmentation Mask Selection* via Top-1 support–query similarity to choose the final mask (Fig. 3).

Below, we introduce the entropy-regularized KL gradient flow, detail the refinement with the segmentation mask decoder and the segmentation mask selection, and conclude with the motivation grounded in a convergence analysis.

3.1. Gradient flow of entropy-regularized KL-divergences

This subsection briefly describes the construction of the gradient flow of entropy-regularized KL-divergences for mask decoder-driven gradient flow. More details can be found in the overview in [21] and discriminator gradient flow [1] that forms the basis of our research.

Let $\mathbf{v}$ be a vector of interest (e.g., a latent vector of a query image), and let $\rho(\mathbf{v})$ and $\mu(\mathbf{v})$ be the candidate and target (ideal) probability density functions (pdf) over $\mathbf{v}$, respectively, assuming that the optimal segmentation mask is distributed over μ in the latent space. For simplicity, we restrict our discussion to pdfs defined on $\mathbf{v}$ that belongs to the 2-Wasserstein space $\mathcal{W}_2$.

We now employ the gradient flow to update $\mathbf{v}$ in order to make the candidate distribution ρ approximate the target distribution μ. Specifically, our focus lies in minimizing the entropy-regularized KL-divergence $\mathrm{F}_\mu(\rho)$ between the target distribution μ and its candidate distribution ρ as:

$$\begin{aligned}\min_{\rho} \mathrm{F}_\mu(\rho) &= \min_{\rho}\Big\{\mathrm{KL}(\mu\|\rho) - \gamma\,\mathrm{H}(\rho)\Big\} \\ &= \min_{\mathbf{v}} \left\{ \begin{array}{l} \displaystyle\int \log\Big(\frac{\mu(\mathbf{v})}{\rho(\mathbf{v})}\Big)\mu(\mathbf{v})\,d\mathbf{v} \\ \quad + \gamma \displaystyle\int \rho(\mathbf{v}) \log\big(\rho(\mathbf{v})\big)\,d\mathbf{v} \end{array} \right\},\end{aligned} \tag{1}$$

where $\mathrm{KL}(\mu\|\rho)$ represents the KL-divergence between two distributions μ and ρ, $\mathrm{H}(\cdot)$ denotes the entropy function, and $\gamma > 0$ is a hyperparmeter that controls the strength of the entropy regularization. Then, the gradient flow of the functional $\mathrm{F}_\mu(\rho)$ is given by:

$$\frac{\partial \rho}{\partial t} = -\nabla \mathrm{F}_\mu(\rho_t), \tag{2}$$

where the subscript t refers to the time (iteration) index throughout this work. Equivalently, Eq. (1) follows a partial differential equation as:

$$\begin{aligned}\partial_t \rho_t(\mathbf{v}) - \nabla_{\mathbf{v}} \cdot (\rho_t(\mathbf{v}) \nabla_{\mathbf{v}} \log(\rho_t(\mathbf{v})/\mu(\mathbf{v}))) \\ - \gamma \Delta_{\mathbf{vv}} \rho_t(\mathbf{v}) = 0,\end{aligned} \tag{3}$$

where $\nabla_{\mathbf{v}}$ and $\Delta_{\mathbf{vv}}$ denote the divergence and the Laplace operators, respectively [1]. The Eq. (3), which is a type of Fokker-Plank equation [20], can be rewritten as an equivalent stochastic differential equation (SDE) defined as:

$$d\mathbf{v}_t = -\nabla_{\mathbf{v}} \log(\rho_t(\mathbf{v})/\mu(\mathbf{v}))dt + \sqrt{2\gamma} d\mathbf{w}_t, \tag{4}$$

where $\mathbf{w}_t$ follows a standard Wiener process [20].

We can simulate a sample $\mathbf{v}_0 \sim \rho_0 = \rho$ using Eq. (4) to obtain a sample close to μ. In practice, this simulation is approximated using the Euler-Maruyama method [12]:

$$\mathbf{v}_{t+1} = \mathbf{v}_t - \eta \nabla_{\mathbf{v}} \log(\rho_t(\mathbf{v})/\mu(\mathbf{v})) + \sqrt{2\gamma\eta}\xi_t, \tag{5}$$

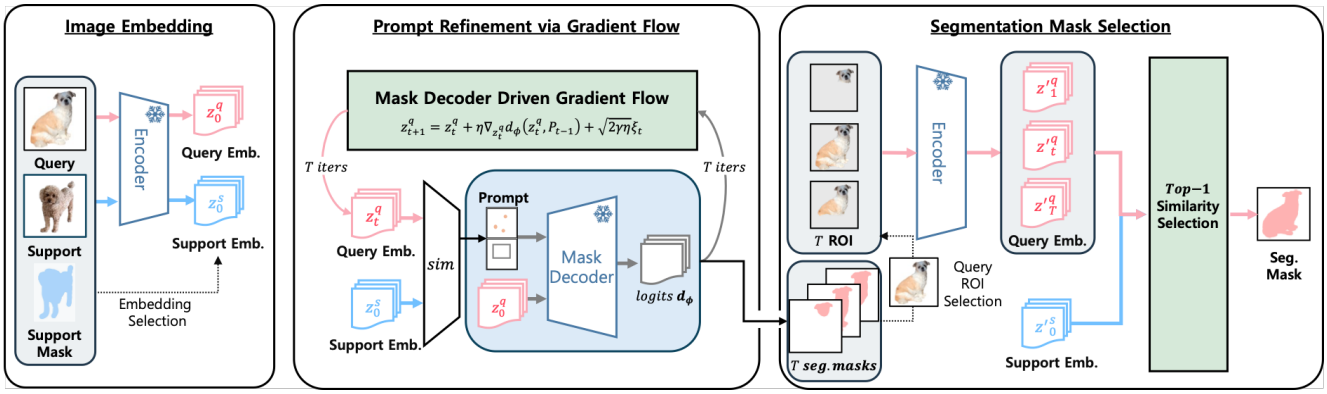


Figure 3. **Overview of PR-MaGIC**. *Encoder* box denotes image encoder (E_θ), *sim* box denotes similarity computing module, and *Mask Decoder* box denotes SAM's mask decoder. SAM's prompt encoder is omitted here to show the clear flow of our method, and green boxes denote our core method. **1) Image Embedding**: *Encoder* maps the support and query images to embeddings $\mathbf{z}_0^s$ and $\mathbf{z}_0^q$, with $\mathbf{z}_0^s$ selected using the given support mask. **2) Prompt Refinement via Gradient Flow**: Using *Mask Decoder*'s logit d_ϕ, we update the query embeddings by the mask decoder driven gradient flow, resample prompts P_{t+1} from S_{t+1}, and iterate for t=0. . .$T-1$ to form a candidate set of masks $\{\hat{\mathbf{m}}_t\}_{t=1}^T$. **3) Segmentation Mask Selection (Top-1 similarity selection)**: Among the generated masks, we select the one whose mask-aware query embedding $z_t'^q$ exhibits the highest similarity to the support embedding z'^s.

where $\eta > 0$ is the step size, $\xi_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $t \in [0, T]$ for the predefined number of iterations T. Furthermore, we can address intractability of $\mu(\mathbf{v})$ in Eq. (5) by approximating the density ratio $\frac{\rho_t(\mathbf{v})}{\mu(\mathbf{v})}$ as $\frac{\rho_0\mathbf{v})}{\mu(\mathbf{v})}$, as $\frac{\rho_t(\mathbf{v})}{\mu(\mathbf{v})} \approx \frac{\rho_0(\mathbf{v})}{\mu(\mathbf{v})}$ for small t and $\eta \to 0$. Given a prediction head $D_\phi(\mathbf{v})$ (e.g., mask decoder parameterized by ϕ) that represents the conditional probability of $\mathbf{v}$ being a sample from μ, the following expression provides an approximation for the density ratio [24]:

$$\frac{\rho_0(\mathbf{v})}{\mu(\mathbf{v})} = \frac{1 - D_\phi(\mathbf{v})}{D_\phi(\mathbf{v})} = \exp\left(-d_\phi(\mathbf{v})\right). \tag{6}$$

Substituting Eq. (6) into Eq. (5) yields the following results:

$$\begin{aligned} \mathbf{v}_{t+1} &= \mathbf{v}_t - \eta\nabla_\mathbf{v} \log\left(\frac{1 - D_\phi(\mathbf{v})}{D_\phi(\mathbf{v})}\right) + \sqrt{2\gamma\eta}\xi_t \\ &= \mathbf{v}_t + \eta\nabla_\mathbf{v} d_\phi(\mathbf{v}) + \sqrt{2\gamma\eta}\xi_t, \end{aligned} \tag{7}$$

where $d_\phi(\mathbf{v}) = -\log((1 - D_\phi)/D_\phi)$ is the logit output of the predictor D_ϕ[1].

In this work, ρ and μ denote the distributions of query embeddings from the VFM encoder and optimal embeddings for the mask decoder, respectively, with D_ϕ representing SAM's mask decoder (a pixel-wise classifier). While a dedicated discriminator would be more principled for D_ϕ, we employ this lightweight proxy to enable a training-free framework.

3.2. Prompt refinement via gradient flow

In this section, we elaborate how gradient flow can be adopted for refining predicted masks. We describe the general framework of few-shot segmentation using promptable segmentation models, and then discuss our method for improving mask quality using mask decoder gradient flow. As illustrated in Fig. 3, the first step is to extract image embeddings of a support image $\mathbf{I}^s$ and a query image $\mathbf{I}^q$ by an image encoder E_θ as

$$\mathbf{z}_0^s = E_\theta(\mathbf{I}^s)\,, \quad \mathbf{z}_0^q = E_\theta(\mathbf{I}^q), \tag{8}$$

where $\mathbf{z}_0^s$ and $\mathbf{z}_0^q$ are initial image embeddings of $\mathbf{I}^s$ and $\mathbf{I}^q$, respectively. Across the two images, Eq. (8) yields two embedding sets, $\{\mathbf{z}_{0,i}^s\}_{i=1}^{N_s}$ and $\{\mathbf{z}_{0,j}^q\}_{j=1}^{N}$, where N_s is the number of pixels within the support mask and N is the total number of pixels in the embedding space. Then, we compute the initial similarity matrix S_0 between the query embeddings and the support embeddings. The i-th row and j-th column of the S_0 is computed as:

$$S_0[i, j] = sim(\mathbf{z}_{0,i}^s, \mathbf{z}_{0,j}^q), \tag{9}$$

where $sim(\cdot, \cdot)$ denotes a similarity function between the two vectors, representing the similarity computation module of the few-shot segmentation model. Finally, we sample prompts P_0 representing the position of the target object, and obtain an initially predicted mask $\hat{\mathbf{m}}_0$ using a mask decoder D_ϕ^{bin}:

$$P_0 = prompt_sampler(S_0), \tag{10}$$

$$\hat{\mathbf{m}}_0 = D_\phi^{bin}(\mathbf{z}_0^q; P_0), \tag{11}$$

where P_0 is a set of prompts obtained from the similarity map S_0, and examples of *prompt_sampler* include top-k sampling [32] and robust sampler [14]. During this process, some prompts in the set P_0 may be false positives or semantically confusing, and several methods have been proposed to address this issue, such as cascaded refinement,

[1]Although the D_ϕ takes prompts such as bounding boxes and prompt points as inputs in addition to $\mathbf{v}$, we omitted them in Eq. (6) and Eq. (7) for clarity.



(a) Source (b) Baseline (c) $t = 1$ (d) $t = 5$ (e) GT

Figure 4. **Refinement process of PR-MaGIC.** Curated illustrations of refinement process with PerSAM-F as the baseline. a) Target objects in the support images with blue masks (bear (top) and airplane (bottom)), b) Segmentation results from PerSAM-F with point prompts in green dots. c), d) Refined prompt and segmentation after one and five iterations, e) Ground truth (G.T.).

mask filtering, and robust sampling [14, 32]. While these approaches yield reasonably refined masks, the results still remain suboptimal (see Fig. 2).

Thus, we propose our process for aligning prompts P_t with the query image semantics z_t^q using the gradient flow derived from the mask decoder using Eq. (7). In particular, we seek to enhance mask quality by iteratively updating the query embeddings, encouraging that predicted masks are more closely aligned with samples from the true mask distribution. Formally, we have the following process in the embedding space:

$$\mathbf{z}_{t+1}^q = \mathbf{z}_t^q + \eta \nabla_{\mathbf{z}_t^q} d_\phi(\mathbf{z}_t^q, P_t) + \sqrt{2\gamma\eta}\xi_t, \tag{12}$$

$$S_{t+1}[i,j] = sim(\mathbf{z}_{0,i}^s, \mathbf{z}_{t+1,j}^q), \tag{13}$$

$$P_{t+1} = prompt_sampler(S_{t+1}), \tag{14}$$

$$\hat{\mathbf{m}}_{t+1} = D_\phi^{bin}(\mathbf{z}_0^q; P_{t+1}). \tag{15}$$

At the t-th iteration, we refine the query embedding $\mathbf{z}_t^q$ via the mask-decoder driven gradient flow in Eq. (12), update the similarity S_{t+1} as in Eq. (13), and resample prompts P_{t+1} from S_{t+1} per Eq. (14). We then decode a candidate segmentation mask $\hat{\mathbf{m}}_{t+1}$ using D_ϕ^{bin} as in Eq. (15). Repeating this procedure for $t = 0, \ldots, T-1$ yields a set of candidates $\mathcal{M} = \{\hat{\mathbf{m}}_t\}_{t=0}^T$.

Restricting the refinement to the prompt space preserves the generality of our method across different visual-prompt frameworks, where feature-level updates could easily depend on architecture-specific representations. Furthermore, the near-optimal assumption of the embedding space does not necessarily hold in practice, and direct modifications of the embedding may introduce instability. To achieve a balanced trade-off between robustness and adaptability, we use the refined embedding z_t^q only for prompt resampling, while decoding is performed with the stable representation z_0^q.

3.3. Segmentation mask selection

For selecting the best segmentation mask, we compute a similarity score s_t between the support embedding and the query embedding aggregated inside $\hat{\mathbf{m}}_t$ (e.g., masked average pooling), and choose the final mask by top-1 similarity.

Let $\mathbf{m}^s \in \{0,1\}^N$ be the given support mask and $\hat{\mathbf{m}}_t \in \{0,1\}^N$ be the candidate mask at iteration t. We obtain embeddings inside a mask, $\mathbf{z}'^s$ and $\mathbf{z}'^q$, using

$$\mathbf{z}'^s = E_\theta(\mathbf{I}^s \odot m^s), \quad \mathbf{z}_t'^q = E_\theta(\mathbf{I}^q \odot \hat{m}_t). \tag{16}$$

We then form the support and query representatives

$$\bar{\mathbf{z}}'^s = \frac{1}{|N|}\sum_{i=1}^N \mathbf{z}'^s, \quad \bar{\mathbf{z}}_t'^q = \frac{1}{|N|}\sum_{i=1}^N \mathbf{z}_t'^q. \tag{17}$$

Given a similarity function $sim(\cdot,\cdot)$, e.g., cosine similarity, we score each candidate by

$$s_t = sim\left(\bar{\mathbf{z}}'^s, \bar{\mathbf{z}}_t'^q\right), \quad t = 0, \ldots, T. \tag{18}$$

Finally, we select the best iteration t^* via top-1 similarity:

$$t^\star = \arg\max_{t\in\{0,\ldots,T\}} s_t. \tag{19}$$

Accordingly, the final mask $\hat{\mathbf{m}}^\star$ is obtained as $\hat{\mathbf{m}}^\star = \hat{\mathbf{m}}_{t^\star}$. Our overall algorithm is summarized in Alg. 1, and Fig. 4 illustrates the progressive refinement process where the points located at the same position are pruned.

Algorithm 1 PR-MaGIC: Gradient-Flow Prompt Refinement with Top-1 Similarity Selection

Require: image encoder E_θ, mask decoder D_ϕ, similarity *sim*, prompt sampler, iterations T
Init: Encode support/query $\rightarrow (\mathbf{z}_0^s, \mathbf{z}_0^q)$; compute S_0; sample P_0.
Initialize candidate masks $\mathcal{M} \leftarrow \emptyset$, scores $\mathcal{S} \leftarrow \emptyset$.
for $t = 0$ to $T-1$ **do**
  Gradient Flow: update query embedding (Eq. (12)).
  Recompute similarity S_{t+1}; resample prompts P_{t+1}.
  Decode mask $\hat{\mathbf{m}}_{t+1} \leftarrow D_\phi^{bin}(\mathbf{z}_0^q; P_{t+1})$.
  Append $\hat{\mathbf{m}}_{t+1}$ to $\mathcal{M}$.
  Score $s_{t+1} \leftarrow sim$(support, query masked by $\hat{\mathbf{m}}_{t+1}$).
  Append s_{t+1} to $\mathcal{S}$.
end for
Best Mask Selection: $t^\star \leftarrow \arg\max s_t$;
return $\hat{\mathbf{m}}^\star \leftarrow \hat{\mathbf{m}}_{t^\star}$

3.4. Motivation for the gradient flow and top-1 similarity selection

Our goal is to refine prompts by updating the query embedding via a gradient flow, and then to select a reliable mask among the candidates generated across iterations. The design is motivated by a convergence perspective.

Motivation for gradient-flow updates. First of all, we demonstrate theoretical convergence of PR-MaGIC in the following proposition, whose proof is given in Appendix A.

Proposition 1. *Let ρ_t be a candidate pdf in 2-Wasserstein space $\mathcal{W}_2$ being updated according to the gradient flow of the entropy-regularized KL-divergence $F_\mu(\rho)$ with local minimum μ^*. If the initial pdf ρ_0 lies in a neighborhood of μ^*, then ρ_t converges to μ^* exponentially as $t \to \infty$.*

Now let ρ_t denote the distribution of query embeddings at iteration t and let $\mu^\star$ be a (local) optimum that is most consistent with the mask decoder. Under the proximity assumption that the initial embedding distribution ρ_0 lies in a neighborhood of $\mu^\star$ (which is a plausible scenario for a pre-trained VFM), the entropy-regularized KL gradient flow drives ρ_t toward $\mu^\star$ at an exponential rate (Proposition 1). Hence, even a small number of iterations improves the mask quality, which justifies our use of gradient flow refinement.

Motivation for top-1 similarity selection. The proximity assumption does not always hold in practice (Sec. 4.4). We empirically observe cases where ρ_0 is not sufficiently close to $\mu^\star$, leading to suboptimal or unstable refinements. To cope with such violations, we record a set of candidate masks over T iterations and introduce a simple yet effective top-1 similarity selection: among the candidates, we choose the mask whose corresponding query embedding is most similar to the support embedding. This selection step acts as a practical safeguard, retaining gains when the assumption holds while improving robustness when it does not (see sensitivity analysis and convergence failure cases in Sec. 4.3 and Sec. 4.4, respectively).

4. Experiments

We evaluate PR-MaGIC across six benchmark datasets for two separate segmentation tasks. All experiments are conducted under identical protocols to assess segmentation quality and generalization.

4.1. Experiment setup

Datasets. We evaluated PR-MaGIC through two tasks: 1) semantic segmentation to assess overall object understanding, and 2) part segmentation to evaluate detailed context understanding. Semantic segmentation performance was assessed on three datasets: **FSS-1000** [13], **LVIS-92^i** [14] and **COCO-20^i** [16]. For part segmentation, we used another three datasets: **PASCAL-Part** [5, 8, 13], **PACO-Part** [14, 19], and **DIS5K** [18]. We strictly followed the data preprocessing and evaluation protocols outlined in [14].

Baselines. We utilize PerSAM-F and Matcher as our baseline methods and demonstrate that PR-MaGIC enhances the segmentation mask quality for both methods. PerSAM-F uses SAM as its backbone model, fully leveraging it to capture visual clues of target objects in the support images. In contrast, Matcher employs DINOv2 [17] with the ViT-L/14 architecture [7] as an image encoder to extract embeddings from both query and support images, and uses SAM's decoder as the segmenter.

Evaluation. To assess the refinement procedure, we report both top-1 selection and oracle mean Intersection over Union (mIoU) gain of PR-MaGIC over baselines. Oracle results are achieved by selecting the most accurate segmentation mask among T=5 refinement steps. The oracle selection is structurally similar to test-time refinement, providing a meaningful upper bound that highlights the capacity of PR-MaGIC. We provide one-shot segmentation performance and additionally provide few-shot (5-shot) results. To further justify our result, we conducted sensitivity analysis on hyperparameters η and T in Sec. 4.3.

Setup of PR-MaGIC. Across all the datasets, we set the number of iterations $T = 5$. The γ for entropy regularization was set as 0.1 without any tuning, and the η for semantic segmentation was 0.001, and that of part segmentation was 0.0001. These η were determined from 10 randomly sampled images from left-aside validation sets of COCO-20^i and PACO-part. The number of points for each method was empirically chosen: For semantic segmentation, Matcher used 8 points and PerSAM-F extracted 5 points as prompts. For part segmentation, the number of points was set to 5 for the Matcher and to 3 for the PerSAM-F, so that they better localize smaller objects. The similarity function $sim()$ for Eq. (14) was adopted according to individual baselines [14, 32]. Finally, the gradients from PR-MaGIC were clipped to ensure stability throughout the overall process. All experiments were conducted with an NVIDIA RTX 6000 ada generation.

4.2. Experiment results

We present experiments on semantic and part segmentation, sensitivity analysis with respect to step size and iteration count, and a sample-wise analysis of optimal step variability. To evaluate the effectiveness of our iterative refinement process, we report three types of results: the baseline **(B)**, PR-MaGIC with Top-1 similarity selection **(T)**, and PR-MaGIC with Oracle selection **(O)**. The oracle provides the best achievable mIoU across all iterations of PR-MaGIC, serving as an empirical upper bound of our model. Due to space limitations, the average mIoU progression with a fixed number of iterations is shown in Appendix B and 1-shot segmentation results across random seeds are provided in Appendix C. Furthermore, qualitative results are provided in Appendix D and E.

4.2.1. Semantic segmentation

The semantic segmentation results evaluate how effectively our refinement process enhances object-level segmentation quality across different datasets. As summarized in Tab. 1, our method consistently improves mIoU on COCO-20^i, FSS-1000, and LVIS-92^i. The PR-MaGIC improves the baseline by +8.8%p on FSS, +2.2%p on COCO, and +2.1%p on LVIS for PerSAM-F, confirming that the refinement process effectively boosts segmentation accuracy at the object level. A small gain observed on FSS-1000 for Matcher(1-shot) is

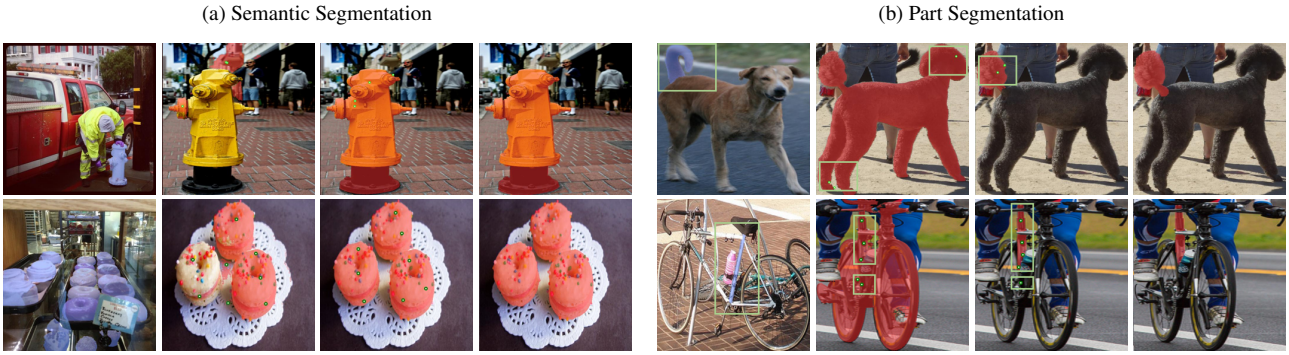


Figure 5. **Qualitative results for semantic and part segmentation.** Baselines from top to bottom: PerSAM-F, Matcher. **a)**: Semantic segmentation. Target objects in support images are highlighted with blue masks, and point prompts are denoted by green dots. **b)**: Part segmentation. Here, target objects are highlighted with blue masks in green boxes, and point prompts are similarly marked and emphasized. Refinement with PR-MaGIC ($3rd$ columns) improves the segmentation over the baselines ($2nd$ columns).

Table 1. mIoU (%) on six datasets. B = baseline, O = PR-MaGIC with Oracle selection (*), T = PR-MaGIC with Top-1 selection. Bold indicates T values that improve over baseline.

	(a) Semantic									**(b) Part / Fine-grained**								
	FSS			**COCO**			**LVIS**			**PACO**			**Pascal**			**DIS**		
Method	*B*	*T*	O^*	*B*	*T*	O^*	*B*	*T*	O^*	*B*	*T*	O^*	*B*	*T*	O^*	*B*	*T*	O^*
PerSAM-F	58.41	**67.19**	72.45	44.64	**46.83**	51.74	42.37	**44.48**	47.29	39.60	**40.72**	43.39	42.72	**43.87**	46.43	46.82	**49.99**	53.46
Matcher(1-shot)	92.08	92.06	93.55	69.53	**71.23**	76.14	59.39	**61.52**	64.75	50.27	**54.08**	56.71	54.76	**58.28**	61.13	46.65	**55.08**	58.10
Matcher(5-shot)	93.26	**93.41**	94.32	67.69	**70.74**	74.88	57.14	**60.79**	63.88	48.66	**53.15**	55.30	54.54	**59.27**	61.55	–	–	–

mainly due to its already saturated baseline performance. In contrast, the refinement yields a pronounced gain for PerSAM-F, showing that our method provides substantial benefits when the initial prediction is less optimal.

Qualitative results in Fig. 5(a) show that baseline predictions often miss object regions or include background noise (e.g., the fire hydrant (1st row) or cupcakes (2nd row)). After refinement, our method captures more precise boundaries and complete object shapes, demonstrating that the proposed refinement process provides consistent improvements without additional training or supervision.

4.2.2. Part segmentation

Part segmentation focuses on assessing whether the refinement process can capture fine-grained object structures and precisely delineate relatively small but semantic parts of objects. Unlike semantic segmentation, which evaluates holistic object understanding, this task emphasizes local boundaries and subtle structural variations within a single object, which is a more sensitive measure of effectiveness of refinement process with PR-MaGIC.

As shown in the right columns of Tab. 1, our method produces consistent and often larger improvements compared to the baseline results. The PR-MaGIC improves the baseline by +3.8%p on PACO and +8.4%p on DIS for Matcher(1-shot), and by +1.1%p, +1.2%p, and +3.2%p on PACO, Pascal, and DIS for PerSAM-F, respectively. These gains indicate improved recovery of fine, part-level regions by our refinement method, especially when baseline predictions lack fine structure or show local fragmentation.

Qualitative results in Fig. 5(b) support these observations. When baseline outputs are misaligned or partially miss object parts (e.g., the wheels or handles of a bicycle (1st row) and the tail of a poodle (2nd row)), the refined predictions yield accurate and continuous part boundaries. Overall, the stronger relative improvement in part and semantic segmentation suggests that the proposed iterative updates preserve local spatial precision while maintaining global consistency.

4.3. Sensitivity analysis

We randomly sampled 100 examples from the set-aside dataset (COCO-20^i training dataset) to conduct the sensitivity analysis on the step size (η) and the number of iterations (T). For the analysis, we evaluated the mIoU over iterations, testing up to 30 iterations with step sizes of 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}. Fig. 6 shows a moving average with a window size of 2 to better identify overall trends.

Findings. The result demonstrates that the proximity assumption $\rho_0 \approx \mu^\star$ is not reliably met in realistic conditions. If the initial embedding distribution ρ_0 were consistently close to the decoder-optimal $\mu^\star$, performance should improve monotonically with small T (Proposition 1). However, our sensitivity results reveal frequent dependency on η and T, and non-monotonic mIoU trajectories. For instance,

- $\eta = 10^{-2}$ yields rapid early gains but degrades with more iterations (instability),

Table 2. Mean ± std of the optimal iterations in PR-MaGIC under Oracle and Top-1 selection across various datasets and baselines.

Model	FSS		COCO		LVIS		PACO-Part		Pascal-Part		DIS	
	Oracle	Top-1	Oracle	Top-1	Oracle	Top-1	Oracle	Top-1	Oracle	Top-1	Oracle	Top-1
PerSAM	2.49 ± 1.74	2.49 ± 1.78	2.42 ± 1.76	2.45 ± 1.76	2.42 ± 1.76	2.44 ± 1.77	2.44 ± 1.78	2.47 ± 1.76	2.36 ± 1.79	2.48 ± 1.76	2.45 ± 1.76	2.51 ± 1.80
Matcher (1shot)	2.45 ± 1.72	2.55 ± 1.71	2.47 ± 1.74	2.47 ± 1.72	2.05 ± 1.80	2.05 ± 1.81	2.44 ± 1.73	2.46 ± 1.73	2.44 ± 1.73	2.42 ± 1.72	2.51 ± 1.73	2.52 ± 1.70
Matcher (5shot)	2.50 ± 1.72	2.50 ± 1.73	2.41 ± 1.71	2.45 ± 1.74	2.29 ± 1.77	2.29 ± 1.77	2.49 ± 1.71	2.48 ± 1.71	2.48 ± 1.72	2.47 ± 1.72	–	–

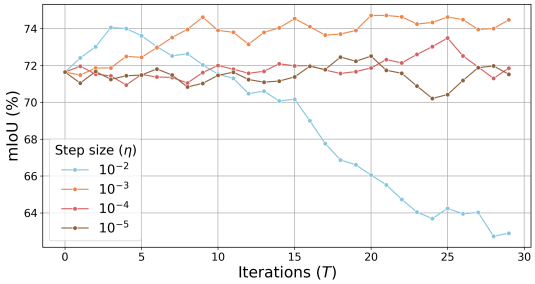


Figure 6. **Sensitivity analysis** on the step size η and iterations T shows that the gradient flow can substantially improve segmentation performance but may exhibit instability under certain conditions, which motivates top-1 mask selection.

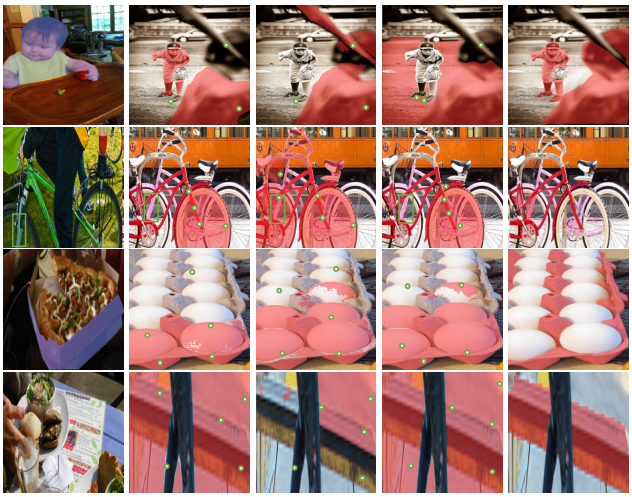
- $\eta \in \{10^{-4}, 10^{-5}\}$ converges slowly and often saturates below the best achievable mIoU (under-refinement).

Implication and design choice. Given this practical difficulty in convergence, mask selection was introduced as in Sec. 3.3. We run the flow for T iterations, collect the mask from each step as a candidate, and apply top-1 support–query similarity to choose the final output. This selection preserves improvements when refinement helps and mitigates degradation when the flow drifts (see Sec. 4.4).

4.4. Sample-wise optimal step variability and convergence challenges

We analyze the refinement process by measuring the average iteration (with corresponding standard deviation) at which the optimal performance was achieved (Tab. 2) for both the oracle and top-1 results. The optimal iteration typically occurred around the third refinement step, suggesting that 5 iterations were sufficient to obtain refined outputs. However, as shown by the relatively large standard deviations, the optimal iteration varied considerably across samples. This variability highlights the necessity of the top-1 selection step, which plays a crucial role in stabilizing and enhancing the overall refinement process.

Limitation. We also highlight challenging scenarios in which PR-MaGIC struggles to converge. When the visual semantics of the support and query images differ substantially (e.g., the 1st and 4th rows of Fig. 7), prompt refinement becomes particularly difficult and often leads to performance degradation. Convergence is also fragile when the visual clues in the support image are ambiguous. As illustrated in Fig. 7, PR-MaGIC fails to localize fine-grained parts of a bicycle (2nd row) and underperforms when the support shows a generic tray while the query target is a specific boundary region (3rd row). These observations collectively emphasize



(a) Support (b) Baseline (c) $t = 1$ (d) $t = 5$ (e) GT

Figure 7. **Convergence failure cases of PR-MaGIC.** Each column represents: (a) Support Image, (b) Baseline output, (c) PR-MaGIC ($t = 1$), (d) PR-MaGIC ($t = 5$), and (e) Ground Truth. The top two rows compare PR-MaGIC with PerSAM-F as the baseline, while the bottom two rows compare PR-MaGIC with Matcher. In this case, PR-MaGIC selects the top-1 mask from the iterations.

that the top-1 selection serves as a practical safeguard against such convergence failures—preserving the fully automatic advantage of PerSAM and Matcher while enhancing robustness and bridging the gap between theoretical convergence and real-world applicability.

5. Conclusion

We present a training-free refinement method that improves segmentation quality using a promptable segmentation model, broadly applicable to auto-prompting frameworks and validated by extensive quantitative and qualitative studies. PR-MaGIC refines the visual clues for SAM's mask decoder via a gradient-flow update in the query embedding space, enabling better alignment with target semantics. Our analysis shows that, under a proximity assumption, the entropy-regularized KL gradient flow yields rapid improvements with few iterations. While sensitivity analyses suggest that our underlying assumption may be fragile in certain cases, the proposed top-1 support–query similarity–based mask selection effectively mitigates this limitation. Importantly, PR-MaGIC remains fully training-free, introducing no learnable parameters, architectural changes, or additional data requirements, yet it achieves consistent improvements in segmentation quality with minimal computational overhead.

Acknowledgment. This research was supported by RS-2025-02216257 (70%), RS-2022-II220290 (20%), and RS-2019-II191906 (AI Graduate Program at POSTECH, 10%).

References

[1] Abdul Fatir Ansari, Ming Liang Ang, and Harold Soh. Refining deep generative models via discriminator gradient flow. In *International Conference on Learning Representations*, 2021. 3

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. 1

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[4] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023. 1, 3

[5] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 6

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1, 6

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 6

[9] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2022. 2

[10] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[12] Peter E Kloeden, Eckhard Platen, Peter E Kloeden, and Eckhard Platen. *Stochastic differential equations*. Springer, 1992. 3

[13] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 2869–2878, 2020. 6

[14] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. In *International Conference on Learning Representations*, 2024. 2, 3, 4, 5, 6

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1

[16] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 622–631, 2019. 6

[17] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 3, 6

[18] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *ECCV*, 2022. 6

[19] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023. 6

[20] H Risken. The fokker-planck equation, 1996. 3

[21] Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7:87–154, 2017. 3

[22] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 2

[23] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021. 1

[24] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012. 4

[25] Yanpeng Sun, Jiahui Chen, Shan Zhang, Xinyu Zhang, Qiang Chen, Gang Zhang, Errui Ding, Jingdong Wang, and Zechao Li. Vrp-sam: Sam with visual reference prompt. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 23565–23574, 2024. 3

[26] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 1

[27] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[28] Junde Wu, Wei Ji, Yuanpei Liu, Huazhu Fu, Min Xu, Yanwu Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023. 1

[29] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022. 1

[30] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 3

[31] Jian-Wei Zhang, Yifan Sun, Yi Yang, and Wei Chen. Feature-proxy transformer for few-shot segmentation. In *Advances in Neural Information Processing Systems*, 2022. 2

[32] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. In *International Conference on Learning Representations*, 2024. 1, 2, 3, 4, 5, 6