Mitigating Pose-Scale Discrepancy Bias and Reforming Multi-Support Reasoning for Few-Shot Semantic Segmentation

Shreya Biswas and Zhaozheng Yin

Department of Computer Science, Stony Brook University, Stony Brook, NY, USA
{shrbiswas, zyin}@cs.stonybrook.edu

Abstract. Few-shot semantic segmentation (FSS) often degrades when support and query images exhibit large pose or scale differences since conventional prototype matching operates in geometry-entangled feature spaces. To mitigate this brittleness, we propose a lightweight encoder-decoder framework that disentangles object representations into a geometry-invariant semantic code and a low-dimensional modulation code capturing instance-specific geometric variation. The reconstruction decoder is trained with equivariance constraints so that geometric information gets encoded by the modulation pathway while preserving semantic consistency in a separate branch. As a result, robustness to pose-scale discrepancies is improved before segmentation without explicit cross-image alignment. To further strengthen supervision from limited supports, we introduce a refinement module that synthesizes lightweight augmented views of each support and performs cyclic ensemble refinement to generate more stable predictions. Predictions from multiple supports are then fused using spatially adaptive reliability weighting, producing cleaner and better-aligned query predictions. Across standard FSS benchmarks, our method consistently improves performance - particularly under large viewpoint changes. Ablations confirm that the disentanglement combined with cyclic ensemble and spatial refinement are critical to the gains. Project Website.

Keywords: Few-shot Segmentation · Pose-Scale Bias · Feature Disentanglement · Multi-Support Refinement

1 Introduction

Deep learning-based semantic segmentation heavily depends on large-scale datasets with dense pixel-level annotations that are often costly and labor-intensive to create. Few-shot segmentation (FSS) [4, 20, 23] addresses this challenge by enabling models trained on base classes to generalize to novel categories with only a few annotated samples. In the standard FSS setting, the model is trained on base classes and evaluated on novel classes using a small set of labeled *support images* with pixel-wise segmentation masks, along with unlabeled *query images*. Prior works have explored a variety of strategies, including prototype-based metric learning [4, 33, 39], conditional and guided networks [10, 21, 25], attention-based

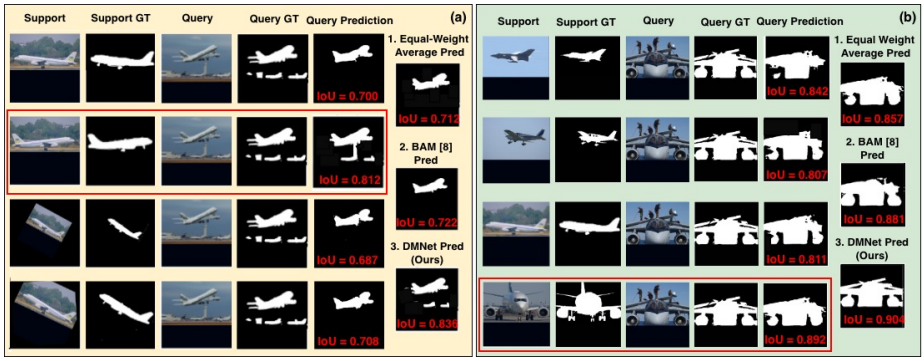


Fig. 1: (a) We demonstrate Pose-Scale Discrepancy Bias in FSS by augmenting the support samples with various similarity transformations. The red box denotes where the FSS model performs the best (usually when the scale and orientation of the support and query objects are aligned well). (b) We show how different support samples yield different predictions - and how different fusion methods lead to various results.

architectures [27,35,36], and latent representation optimization [22,26,40]. Some works go beyond the limited visual cues and incorporate external knowledge sources such as semantic text embeddings [28,41] and knowledge from pretrained foundation models [11, 42].

A critical yet underexplored challenge in FSS is the *Pose-Scale Discrepancy Bias*. Although support and query objects belong to the same category, pronounced differences in their orientation or size can impair correspondence when matching is conducted in a feature space, where semantic and instance-level variations are entangled. Since FSS relies on limited support samples and similarity-based guidance, its performance depends heavily on the stability of feature correspondence between support and query images. Although CNN models exhibit some degree of transformation tolerance [2, 9], they are primarily optimized for category discrimination rather than structured correspondence. Consequently, prototype-based similarity does not explicitly guarantee consistent responses under substantial pose-scale variation. Attention-based FSS methods alleviate some limitations of static prototypes by enabling dense cross-image interactions. However, attention weights are still learned based on feature similarity without explicitly modeling geometric consistency. As a result, both approaches may produce degraded similarity maps when support and query objects exhibit significant pose-scale discrepancies. In few-shot settings, such mismatch can weaken object correspondence and amplify bias toward appearance-specific cues, resulting in fragmented activations and unstable mask predictions.

To empirically demonstrate this bias, we apply controlled transformations such as scaling and rotation to the support image relative to the query. Figure 1a shows that FSS performance is highest when the support and query objects share similar orientation and scale, and it degrades noticeably as the discrepancy increases. Through these synthetic similarity transformations, we reveal that existing FSS models tend to overfit to specific orientations and object sizes,

struggling to generalize across intra-class geometric changes. Additional examples are provided in the Supplementary Material.

A second issue in FSS is the lack of effective aggregation of information from the scarce support samples. Due to limited supervision, the optimization landscape is uncertain, and due to pose-scale bias, not all samples are equally informative (Figure 1b). In conventional K-shot FSS, the most common fusion strategy aggregates information from multiple support examples by uniformly averaging their prototypes or feature embeddings before guiding query segmentation. Although simple and computationally efficient, this approach relies on the strong assumption that all supports are equally informative and relevant to the query. This assumption is rarely valid in practice, leading to inferior performance that may be even worse than the single support with the best alignment (Figure 1[a1, b1]). Support objects may exhibit large variations in pose and scale relative to the query, along with changes in illumination, occlusion, and background clutter. These tend to generate incomplete segmentation predictions.

BAM [8] addresses the above issue by introducing an adjustment factor Ψ, which quantifies the scene difference between support and query pairs using the Frobenius norm of Gram matrix differences computed from low-level feature maps and uses this to fuse support predictions. BAM (Figure 1[a2, b2]) yields better results compared to average fusion. However, Ψ can be dominated by irrelevant scene context rather than by the target object itself. Therefore, a better fusion strategy that identifies the contribution of each support object is essential to achieve robust multi-shot performance (ours in Figure 1[a3, b3]).

To address the above challenges, we propose a novel framework, *Decoupled Matching Network* (DMNet), that explicitly mitigates pose-scale discrepancy bias and improves multi-shot reasoning through three key contributions:

1. **Geometry-Decoupled Disentangled Autoencoder (DAE):** We propose a representation learning framework that decomposes object features into two compact latent codes: a *Semantic Code* and a *Modulation Code*. The *Semantic Code* captures geometry-invariant class identity, while the *Modulation Code* models instance-specific geometric variation. The disentanglement results in semantic matching being performed in a space where pose-scale variation does not interfere. This structured reconstruction reduces mismatched activations and improves object correspondence, ultimately yielding more accurate and stable segmentation predictions.
2. **Cyclic Ensemble Refinement:** To effectively leverage multiple supports, we propose a refinement that uses query-support consistency. We enhance the mutual relation between support and query in a cyclic manner - predicting and refining their segmentation iteratively. This cyclic refinement allows the network to refine query predictions by treating them as pseudo-supports.
3. **Spatially Adaptive Fusion:** We introduce a mechanism that dynamically weights each support's contribution to the query at the pixel level. This weighted aggregation yields more reliable segmentation masks.

2 Related Works

2.1 Pose-Scale Variation in Few-Shot Segmentation

Recent FSS advances increasingly target the support-query mismatch that breaks naive similarity-based prior maps, evolving from attentive K-shot fusion [36] to stronger correspondence models [25]. Transformer-based cross-attention methods [37] have been used to explicitly filter out harmful support regions. But none of these effectively resolve the pose-scale discrepancy bias problem. A broader analysis [1] identifies this bias as a persistent open challenge in FSS, noting that existing methods largely assume spatial consistency between support and query.

PMMs [33] represent one of the earliest efforts to handle intra-class variation in FSS. Instead of constructing a single global prototype per class, they introduce a mixture of multiple local prototypes learned through an expectation-maximization process, capturing diverse visual modes. However, the method remains appearance-driven; it clusters features in embedding space without taking explicit viewpoint changes into consideration. SSFSS [5] introduces a self-guided mechanism in which the query image generates its own "self-support" features to complement the support cues. This design alleviates over-dependence on mismatched supports and enhances adaptability under large appearance shifts. However, the self-support process is unsupervised and purely feature-similarity based.

These methods primarily address mismatch by reweighting or refining feature correspondences, without explicitly structuring the representation to separate semantic identity from instance-level transformation factors. As a result, while they can suppress certain appearance inconsistencies, they lack a dedicated mechanism to model and absorb substantial pose or scale variation. This limitation motivates the need for FSS frameworks that explicitly disentangle transformation-related variation within the representation itself, rather than relying solely on similarity re-calibration.

2.2 Aggregation of Multiple Supports in FSS

Existing multi-support fusion strategies exhibit important limitations at the algorithmic level. PANet [29] enforces a single round of prototype alignment by projecting query predictions back onto the support and encouraging consistency at the prototype level. As a result, the process may propagate errors if initial predictions are imperfect, since there is no explicit mechanism to evaluate or filter unreliable support contributions. BAM [8] emphasizes the suppression of misleading regions by estimating scene-level discrepancy between support and query features. However, the adjustment mechanism is primarily global and feature-statistics driven which may be dominated by background variations rather than object-specific reliability. As a result, supports that are globally similar may still introduce local errors, while locally reliable supports may be underweighted due to scene-level mismatch. DCAMA [24] performs attention-based aggregation by computing dense cross-attention between support and query features. Although attention weights are adaptive, they are implicitly determined by feature similarity and learned attention scores, making the fusion heavily appearance-driven.

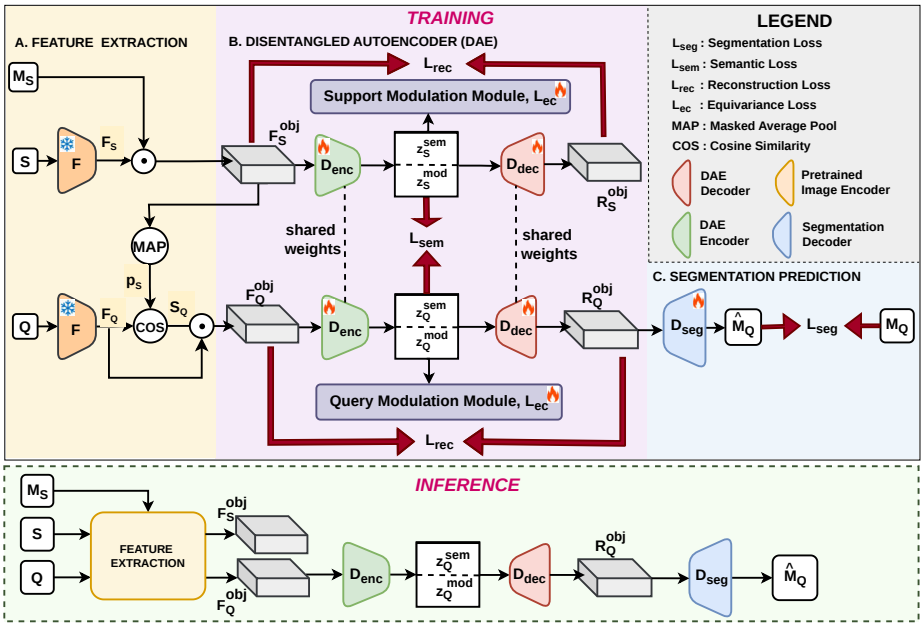


Fig. 2: Geometry-Decoupled Disentangled Autoencoder (DAE) in DMNet.

3 Methodology

3.1 Problem Setting

A FSS task involves two disjoint datasets: a *training set* D_{train} containing base classes C_{base}, and a *testing set* D_{test} comprising novel classes C_{novel} ($C_{\text{base}} \cap C_{\text{novel}} = \varnothing$). The objective is to learn transferable segmentation knowledge from D_{train} that generalizes to D_{test} with very few labeled samples (*support*) from novel categories. FSS is typically trained in an *episodic manner*, mimicking few-shot conditions during both training and testing. Each episode consists of:

- **Support Set** (S): A set of labeled samples ($S = \{(I_S^k, M_S^k)\}_{k=1}^K$) for a target class c, where I_S^k and M_S^k denote the k-th support image and its corresponding binary mask, respectively. We call this K-*shot* scenario.
- **Query Image** (Q): An unlabeled image I_Q containing an object of the same class as S; its ground-truth mask M_Q.

During *training*, both S and Q are sampled from D_{train}, and M_Q is used to compute the segmentation loss. During *testing*, episodes are sampled from D_{test}, and the model predicts the segmentation mask ($\widehat{M_Q}$) for I_Q.

3.2 Method Overview

A straightforward way to solve pose-scale bias would be to explicitly align support and query images, e.g., by estimating a geometric transform and warping

one image to match the other. However, in FSS each episode involves unseen categories and varying object instances, making reliable correspondence estimation at test time extremely challenging. Classical geometric matching (e.g., keypoints + RANSAC [13]) is brittle on deformable objects and severe clutter, while STN-style modules [7] still depend on strong correspondence cues, and often overfit to dominant geometric modes observed during training.

Instead of cross-image alignment, we adopt a different perspective: *geometric (pose-scale) variation should not disrupt semantic matching*. The proposed DM-Net (Figure 2) addresses this by disentangling semantics from instance-specific geometric variation. We encode (D_{enc}) the object-centric feature maps of the support and query (F_S^{obj} and F_Q^{obj}) as a combination of two latent factors: (i) a geometry-invariant *Semantic Code* (z_S^{sem} and z_Q^{sem}) capturing class identity, and (ii) a *Modulation Code* (z_S^{mod} and z_Q^{mod}) encoding instance-dependent geometric variation. To enforce this, we design modulation modules to learn a low-dimensional subspace for the support and query features (shown in purple in Figure 2), in which a 2D vector - termed the *geometric pointer* - parameterizes and visualizes the transformation in the 2D space. A reconstruction decoder (D_{dec}) recombines the codes and produces reconstructed object features (R_Q^{obj} and R_S^{obj}). A segmentation decoder (D_{seg}) then provides the query prediction ($\widehat{M}_Q$). This isolates instance-specific variation within the modulation pathway, enabling semantic correspondence to be established in a disentangled space.

3.3 Model Architecture

(A) Feature Extraction Given query Q and support S with its binary mask M_S, we first extract feature maps ($F_S, F_Q \in \mathbb{R}^{C\times H\times W}$, where C is the channel dimension, and H and W are the height and width of the feature volume) using a shared backbone network F. We obtain the support object feature ($F_S^{obj} \in \mathbb{R}^{C\times H\times W}$) and its compact prototype ($p_S \in \mathbb{R}^C$) by:

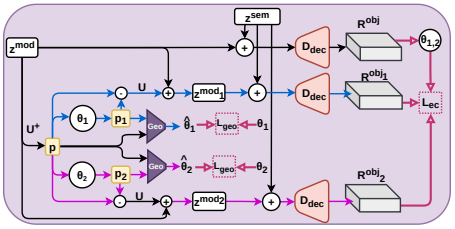
$$F_S^{obj} = F_S \odot M_S, \qquad p_S = \mathrm{MAP}(F_S^{obj}), \tag{1}$$

where $\odot$ denotes channel-wise element-wise multiplication and MAP is Masked Average Pooling. To compute the query object feature ($F_Q^{obj} \in \mathbb{R}^{C\times H\times W}$), we compute a pixel-wise cosine similarity ($S_Q \in \mathbb{R}^{H\times W}$) between F_Q and p_S for each spatial location (x, y):

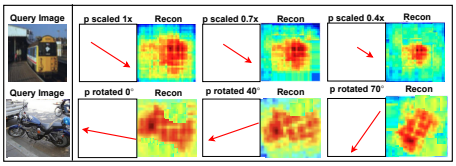
$$S_Q(x,y) = \frac{F_Q(x,y)^T p_S}{\|F_Q(x,y)\|_2 \, \|p_S\|_2}, \qquad F_Q^{obj} = F_Q \odot S_Q. \tag{2}$$

(B) Geometry-Decoupled Disentangled Autoencoder (DAE) For both support and query branches, F_S^{obj} and F_Q^{obj} are passed through a shared encoder D_{enc}, producing a compact latent representation $\in \mathbb{R}^{2\times d}$. This latent vector is partitioned into two components: Semantic Codes ($z_S^{\mathrm{sem}}, z_Q^{\mathrm{sem}} \in \mathbb{R}^d$) and Modulation Codes ($z_S^{\mathrm{mod}}, z_Q^{\mathrm{mod}} \in \mathbb{R}^d$) respectively. D_{dec} is designed to reconstruct the object features from the respective latent codes:

$$R_S^{\mathrm{obj}} = D_{dec}(z_S^{sem} \oplus z_S^{mod}), \qquad R_Q^{\mathrm{obj}} = D_{dec}(z_Q^{sem} \oplus z_Q^{mod}), \tag{3}$$



(a) Modulation Module.



(b) Effect of applying similarity transforms on the geometric pointer p on reconstructed features.

Fig. 3: (a) Modulation Module. (b) Geometric pointer p.

where $\oplus$ represents the channel-wise concatenation operation. We add a reconstruction loss $\mathcal{L}_{\text{rec}}$ to keep D_{dec}'s outputs $(R_S^{\text{obj}}, R_Q^{\text{obj}})$ on the same feature manifold as F_S^{obj} and F_Q^{obj}, preventing implausible reconstructed features (Figure 9).

$$\mathcal{L}_{\text{rec}} = \|R_S^{\text{obj}} - F_S^{\text{obj}}\|_2^2 + \|R_Q^{\text{obj}} - F_Q^{\text{obj}}\|_2^2. \tag{4}$$

Semantic Consistency: To ensure z_S^{sem} and z_Q^{sem} encode the common (class-level) features across the support and query branches, we minimize:

$$L_{sem} = \|z_S^{\text{sem}} - z_Q^{\text{sem}}\|_2^2. \tag{5}$$

Modulation Module: We project z_Q^{mod} and z_S^{mod} [1] onto a 2-D subspace via a learnable matrix $U \in \mathbb{R}^{d\times 2}$, resulting in their *geometric pointer* p:

$$p \;=\; U^+ z^{\text{mod}} \in \mathbb{R}^{2\times 1}, \qquad U^+ = (U^\top U + \lambda I)^{-1} U^\top, \lambda > 0. \tag{6}$$

Our idea is to make p a representative of the pose-scale of the corresponding image, so that any similarity transformation (SIM) applied to it is reflected in the reconstructed features. So, we randomly sample two sets of transformation parameters $(\theta_i = (t_{xi}, t_{yi}, s_i, \phi_i),\ i \in \{1, 2\})$ where t_{xi} and t_{yi} are translation components, s_i represents scaling factor, and ϕ_i denotes rotation angle. We apply SIM_{θ_i} transformations on p to generate two versions of p (p_i).

$$p_i = s_i\, Rot(\phi_i)\, p + t_i, \quad Rot(\phi_i) = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix}, \quad t_i = \begin{bmatrix} t_{xi} \\ t_{yi} \end{bmatrix}, \tag{7}$$

where $Rot(\phi_i)$ is the rotation matrix. Using multiple transformed versions discourages the network from learning a shortcut that directly predicts the transformation parameters without encoding meaningful geometric structure in the modulation space. To ensure that geometric variations encoded in the pointer are faithfully reflected in the reconstructed features, we enforce a consistent relationship between transformations in the modulation space and changes in the decoded feature maps. Specifically, when the pointer is transformed from p to p_i, the corresponding reconstructed feature should undergo the same transformation. For this, we convert p_i back to $\mathbb{R}^D$ space to generate z^{mod_i}. Modulating

[1] For simplicity, in Figure 3a and description below, we drop the subscript - the Modulation Module is applied to both support and query branches

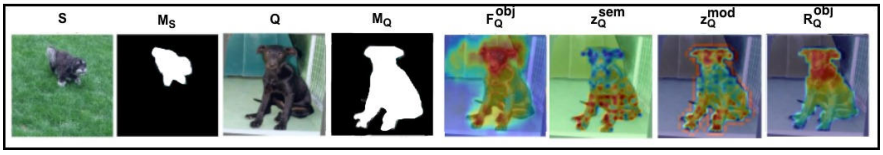


Fig. 4: The features in DAE.

from p to p_i produces a delta change $p_i - p$. The matrix U maps back $(p_i - p)$ into the latent space, so the updated modulation code z^{mod_i} is:

$$z^{\mathrm{mod}_i} = z^{\mathrm{mod}} + U\,(p_i - p). \tag{8}$$

The decoder D_{dec} then reconstructs the transformed object features:

$$R^{\mathrm{obj}_i} = D_{dec}(\, z^{\mathrm{sem}} \oplus z^{\mathrm{mod}_i}), \quad i \in \{1, 2\}. \tag{9}$$

Specifically, when a controlled similarity transformation is applied to p, the reconstructed object feature should undergo the same transformation in spatial space to ensure geometric variations are consistently modeled within the modulation pathway rather than entangled with semantic identity. This is enforced by an equivariance consistency loss $\mathcal{L}_{\mathrm{ec}}$:

$$\mathcal{L}_{\mathrm{ec}} = \sum_{i\in\{1,2\}} \|R^{\mathrm{obj}_i} - \mathrm{SIM}_{\theta_i}(R^{\mathrm{obj}})\|_2^2 = \sum_{i\in\{1,2\}} \|D_{dec}(z^{\mathrm{sem}} \oplus z^{\mathrm{mod}_i}) - \mathrm{SIM}_{\theta_i}\,(D_{dec}(z^{\mathrm{sem}} \oplus z^{\mathrm{mod}}))\|_2^2. \tag{10}$$

To keep p from encoding noise, a *Geo Head* (a multi-layer perceptron) predicts $\hat{\theta}_i$ from (p, p_i) and compares it to the applied transform θ_i by minimizing:

$$\mathcal{L}_{\mathrm{geo}} = \sum_{i\in\{1,2\}} \left\|\hat{\theta}_i - \theta_i\right\|_2^2. \tag{11}$$

Figure 3a shows the overall idea of the Modulation Module. Figure 3b shows two examples of reconstructed features R^{obj_i} when we apply particular SIM_{θ_i} transformations on the geometric pointer p. As we see, controlled changes on p cause similarly structured transformations in the decoded feature map. This shows that the model learns an internal alignment mechanism: by learning how changes in the geometric pointer p affect feature reconstruction during training, the decoder becomes capable of generating geometry-consistent features that align with the query configuration during inference.

Figure 4 shows the DAE feature representations in an example of a significant pose-scale difference - the support object is much smaller than the query object and has a large pose variation. We observe that F_Q^{obj} has strong responses on the object but also, noisy activations in background under such variations. z_Q^{sem} captures class-level semantics (background activations are weaker and it focuses on the object identity), while z_Q^{mod} captures instance-specific geometric features related to the object's configuration. The reconstructed feature R_Q^{obj} from D_{dec}, combining both codes, is cleaner and with stronger activation over the full dog, demonstrating that disentangled latent codes provide complementary guidance (semantic stability and geometry-specific refinement) for mask prediction.

(C) Segmentation Map Prediction Segmentation is a pixel-wise classification into object and background. A segmentation decoder D_{seg} (architecture details in Section 4) is designed to predict the per-pixel object probability map P for query image I_Q and its binary prediction mask $\widehat{M}_Q$:

$$\widehat{M}_Q(x) = \mathbf{1}[P_Q(x) \geq \tau], \quad \tau \text{ is the threshold,} \tag{12}$$

where $\mathbf{1}[.]$ is an indicator function, thresholding the probability at each pixel x by τ. Segmentation is optimized by minimizing the Cross-Entropy Loss (CE):

$$\mathcal{L}_{seg} = \mathrm{CE}(P_Q, M_Q). \tag{13}$$

Total Training Loss: Training uses multiple complementary objectives:

$$\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{seg}} + \lambda_{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}} + \lambda_{\mathrm{ec}}\mathcal{L}_{\mathrm{ec}} + \lambda_{\mathrm{geo}}\mathcal{L}_{\mathrm{geo}}, \tag{14}$$

where λ_{sem}, λ_{ec}, λ_{geo} are respective loss weights.

Inference-time Role of DAE During inference, no synthetic transformations are applied. D_{enc} extracts object features and encodes query object features into semantic and modulation codes, and D_{dec} reconstructs refined query object features by recombining them. As p is used to enforce an equivariance constraint in the modulation space during training, this teaches the decoder how geometric variations affect object features, enabling it to reconstruct geometry-consistent representations during inference without explicitly applying transformations before being passed to D_{seg}.

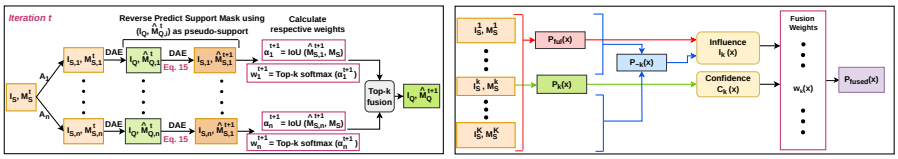
3.4 Query Prediction Refinement

While the DAE yields an initial predicted query mask $\widehat{M}_Q$ (for iteration $t = 0$, we will refer to it as $\widehat{M}_Q^{(0)}$), boundary imprecision and part misalignment can persist, especially under large pose-scale gaps. We therefore apply a light refinement stage that (i) enforces a *cyclic-ensemble* across support augmentations and (ii) fuses multi-support predictions using *spatially adaptive* reliability maps.

Cyclic Ensemble Refinement As we have shown in Figure 1, FSS models are sensitive to different support augmentations - and often the support object which matches the query's orientation-scale yields the best performance. To exploit this, at iteration t, we generate multiple augmentations of (I_S, M_S) by applying n random augmentations A_i (scaling, flipping, rotating) to generate $(I_{S,i}, M_{S,i}^t)_{i=1}^n$. We feed these augmented pairs to the DAE to generate the predicted query mask at iteration t ($\widehat{M}_Q^t$). $\widehat{M}_Q^t$ now acts as the pseudo-support and we *reverse-predict* the supports' prediction masks for iteration $t+1$:

$$\widehat{M}_{S,i}^{t+1} \;=\; \mathrm{DAE}\big(I_{S,i} \;\big|\; (I_Q, \widehat{M}_Q^t)\big). \tag{15}$$

We now compare how close the predicted masks for the support samples are to the ground truth by computing their compatibility scores:



(a) Cyclic Ensemble Refinement Mechanism. **(b)** Spatially Adaptive Fusion Mechanism.

Fig. 5: Refinement Modules.

$$\alpha_i^{t+1} \;=\; \mathrm{IoU}\big(\widehat{M}_{S,i}^{\,t+1}, M_S\big). \tag{16}$$

Top-K augmentations per support by their respective α_i are selected, and from each of these selected supports, we generate a fused query prediction:

$$\widehat{M}_Q^{t+1} = \mathbf{1}\left[\left(\sum_{i\in\text{Top-k}} \omega_i^{t+1}\,\mathrm{DAE}\Big(I_Q \,|\, (I_{S,i},\, \widehat{M}_{S,i}^{\,t+1})\Big)\right) \geq \tau\right], \quad \text{where } \omega_i^{t+1} \;=\; \frac{\exp(\alpha_i^{t+1})}{\sum_{j\in\text{Top-K}} \exp(\alpha_j^{t+1})}. \tag{17}$$

The iteration of this cyclic ensemble refinement is summarized in Figure 5a. This process continues until convergence (e.g., $\|\widehat{M}_Q^{t+1} - \widehat{M}_Q^{t}\| < \varepsilon$) or a fixed number of iterations T (We have chosen $\varepsilon = 10^{-4}, T = 3$). A good initial query prediction will act as a reliable pseudo-support and yield a strong prediction for the support. Else, there is scope to refine the predictions in further iterations.

Spatially Adaptive Fusion Even after implementing the Disentangled Autoencoder architecture and Cyclic Ensemble Refinement, some cases exhibit extreme pose-scale differences (Figure 6). Also, from Figure 1, we know that not all support sample regions have equal importance for the query predictions. For this, we define pixel-wise spatial weights $w_k(x)$ (where x represents each pixel in the query) that combine the *influence* the k-th support sample (I_S^k) has for the prediction, as well as the *confidence* with which it makes those predictions.

Influence: Let $P_k(x)$ denote the one-shot prediction from I_S^k for pixel x. The *influence* of the k-th support on pixel x is defined as:

$$I_k(x) \;=\; \max\Big(0,\; P_{full}(x) - P_{-k}(x)\Big), \tag{18}$$

where $P_{full}(x)$ is the mean of object probabilities over all K supports. $P_{-k}(x)$ is computed by averaging over all supports excluding the k-th one.

Confidence: This metric is used to denote how certain I_S^k's predictions are:

$$C_k(x) \;=\; 1 \;-\; H\big(P_k(x)\big), \tag{19}$$

where H is the binary entropy function computed using base-2 logarithm, allowing $C_k(x) \in [0, 1]$. Final spatial weights $w_k(x)$ are obtained by normalizing the combined scores and then we calculate the final fused query probability as:

$$w_k(x) \;=\; \frac{\exp\big(I_k(x)\, C_k(x)\big)}{\sum_{j=1}^{K} \exp\big(I_j(x)\, C_j(x)\big)}, \qquad P_{\text{fused}}(x) \;=\; \sum_{k=1}^{K} w_k(x)\, P_k(x). \tag{20}$$

This spatially adaptive fusion (summarized in Figure 5b) enables the model to exploit complementary support regions while down-weighting misleading cues, resulting in sharper and more reliable query segmentation.

Table 1: Experimental results on PASCAL-5^i and COCO-20^i. F denotes the pretrained image feature encoder. S1-S4 denote the dataset splits. mIoU is the evaluation metric. **Bold** means best performance, underline means second best performance.

Method	F	PASCAL-5^i (1-shot)					PASCAL-5^i (5-shot)					COCO-20^i (1-shot)					COCO-20^i (5-shot)				
		S1	S2	S3	S4	Mean	S1	S2	S3	S4	Mean	S1	S2	S3	S4	Mean	S1	S2	S3	S4	Mean
PANet [29]	VGG-16	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7	-	-	-	-	20.9	-	-	-	-	29.7
RePRI [33]	VGG-16	47.1	65.8	50.6	48.5	53.0	50.0	66.46	51.94	47.64	54.0	-	-	-	-	-	-	-	-	-	-
PFENet [25]	VGG-16	56.9	68.2	54.4	52.4	58.0	59.0	69.1	54.8	52.9	59.0	33.4	36.0	34.1	32.8	34.1	35.9	40.7	38.1	36.1	37.7
HSNet [14]	VGG-16	59.6	65.7	59.6	54.0	59.7	64.9	69.0	64.1	58.6	64.1 -	-	-	-	-	-	-	-	-	-	-
BAM [8]	VGG-16	63.2	70.8	66.1	57.5	64.4	67.4	73.1	70.6	64.0	68.8	38.9	47.0	46.4	41.6	43.5	47.0	52.6	48.6	49.1	49.3
HDMNet [18]	VGG-16	64.8	71.4	67.7	56.4	65.1	68.1	73.1	71.8	64.0	69.3	40.7	50.6	48.2	44.0	45.9	47.0	56.5	54.1	51.9	52.4
SCCAN [32]	VGG-16	62.9	70.7	64.8	54.3	63.1	66.1	73.4	68.3	63.1	67.7	38.9	46.3	45.5	42.8	43.4	46.4	51.7	53.5	48.0	49.9
AENet [30]	VGG-16	66.3	73.3	68.5	58.4	66.6	70.8	75.1	72.2	64.2	70.6	40.3	50.4	47.9	44.9	45.9	45.8	56.3	55.8	53.4	52.8
HMNet [31]	VGG-16	66.7	74.5	68.9	59.0	67.3	70.5	76.0	72.2	65.7	71.1	44.2	51.8	51.9	48.4	49.1	48.8	58.0	57.9	53.2	54.5
DeFSS [19]	VGG-16	67.5	76.4	69.8	60.7	68.6	71.5	76.8	72.7	66.9	72.0	45.8	52.6	53.0	49.9	50.3	49.5	59.2	58.3	55.0	55.5
DMNet (Ours)	VGG-16	**69.6**	**78.2**	**71.8**	**63.0**	**70.7**	**73.3**	**77.7**	**74.9**	**69.8**	**74.0**	**47.5**	**54.1**	**55.0**	**52.4**	**52.3**	**51.7**	**61.2**	**60.8**	**57.8**	**57.9**
HSNet [14]	ResNet-50	64.3	70.7	60.3	60.5	64.0	70.3	73.2	67.4	67.1	69.5	36.3	43.1	38.7	38.7	39.2	43.3	51.3	48.2	45.0	46.9
CyCTR [38]	ResNet-50	65.7	71.0	59.5	59.7	64.0	69.3	73.5	63.8	63.5	67.5	38.9	43.0	39.6	39.8	40.3	41.1	48.9	45.2	47.0	45.6
SSFSS [5]	ResNet-50	60.5	67.8	66.4	51.0	61.4	67.5	72.3	75.2	62.1	69.3	35.5	39.0	37.9	36.7	37.4	40.6	47.0	45.1	43.9	44.1
DCAMA [24]	ResNet-50	67.5	72.3	59.6	59.0	64.6	70.5	73.9	63.7	65.8	68.5	41.9	45.1	44.4	41.7	43.3	45.9	50.5	50.7	46.0	48.3
BAM [8]	ResNet-50	69.0	73.6	67.6	61.1	67.8	70.6	75.1	70.8	67.2	70.9	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
HDMNet [18]	ResNet-50	71.0	75.4	68.9	62.1	69.4	71.3	76.2	71.3	68.5	71.8	43.8	55.3	51.6	49.4	50.0	50.6	61.6	55.7	56.0	56.0
SCCAN [32]	ResNet-50	68.3	72.5	66.8	59.8	66.8	72.3	74.1	69.1	65.6	70.3	40.4	49.7	49.6	45.6	46.3	47.2	57.2	59.2	52.1	53.9
AENet [30]	ResNet-50	71.3	75.9	68.6	65.4	70.3	73.9	77.8	73.3	72.0	74.2	45.4	57.1	52.6	50.0	51.3	52.7	62.6	56.8	56.1	57.1
I^2R [6]	ResNet-50	71.8	77.1	71.5	68.4	72.2	72.6	78.9	74.7	72.6	74.7	47.2	56.9	52.7	51.7	52.1	51.3	60.3	58.5	52.8	55.7
HMNet [31]	ResNet-50	72.2	75.4	70.0	63.9	70.4	74.2	77.3	74.1	70.9	74.1	45.5	58.7	52.9	51.4	52.1	53.4	64.6	60.8	56.8	58.9
DeFSS [19]	ResNet-50	73.7	76.6	70.9	65.8	71.7	75.0	78.6	74.7	72.2	75.1	46.5	59.7	54.8	53.2	53.7	55.1	66.0	61.3	57.5	60.0
DMNet (Ours)	ResNet-50	**76.1**	**81.9**	**75.0**	**69.3**	**75.6**	**77.4**	**79.9**	**76.7**	**74.8**	**77.5**	**50.7**	**64.1**	**57.7**	**57.4**	**57.5**	**58.0**	**69.4**	**65.1**	**59.8**	**63.1**

4 Experiments

Datasets. PASCAL-5^i [23] comprises 20 distinct classes. COCO-20^i [15] presents a larger dataset with 80 classes. Following the protocols used by [25], during training, both PASCAL-5^i and COCO-20^i are partitioned into four folds for cross-validation purposes, with each fold containing either 5 and 20 classes respectively. Within each fold, the training set is the union of the other three folds, while the fold itself is reserved for testing. During testing, we randomly sample 1,000 episodes from PASCAL-5^i and 4,000 episodes from COCO-20^i.

Implementation Details. For the backbone feature extractor F, we take frozen ImageNet pretrained VGG and ResNet50, for fair comparisons with many FSS methods. D_{enc} consists of 3×3 convolutional layers with ReLU to downsample the feature to $2 \times d$ ($d = 32$), which is then split into z^{sem} and z^{mod} (dimension d each). D_{dec} mirrors D_{enc} but uses upsampling blocks for reconstruction. D_{seg} is a simple convolutional module followed by a classification head with two output classes to yield the segmentation prediction. The classification head is composed of one 3×3 convolution and 1×1 convolution with the Softmax function.

DMNet is implemented using PyTorch [17] using one NVIDIA GPU (RTX 4090). The training follows an episodic learning setup, running for 50 epochs on COCO-20^i and 100 epochs on PASCAL-5^i. Batch size B is set to 2 for COCO-20^i and 8 for PASCAL-5^i. During training, we employ the AdamW optimizer [12], with a learning rate of 0.0001 and a weight decay of 0.01. We set the loss weights: $\lambda_{sem} = 0.2$, $\lambda_{ec} = 0.5$, and $\lambda_{geo} = 0.1$ using cross-validation with a 90-10 split in each fold. We use mean intersection over union (mIoU) as the evaluation metric.

5 Results

Quantitative Analysis. Table 1 reports mIoU comparisons on PASCAL-5^i and COCO-20^i under the standard 1-shot and 5-shot settings. We compare DMNet

Table 2: Comparison with Foundation model-based FSS architectures.

Method	FE	PASCAL-5^i (1-shot)	PASCAL-5^i (5-shot)	COCO-20^i (1-shot)	COCO-20^i (5-shot)	Avg mIoU
SANSA [3]	SAM2-L	–	–	60.2	64.3	62.3
FCP [16]	SAM	73.2	74.0	52.5	58.0	64.4
GF-SAM [34]	SAM	72.1	82.6	58.7	66.8	70.1
DMNet	ResNet-50	75.6	77.5	57.5	63.1	68.4
DMNet-SAM	SAM	77.7	79.8	59.2	66.4	70.8

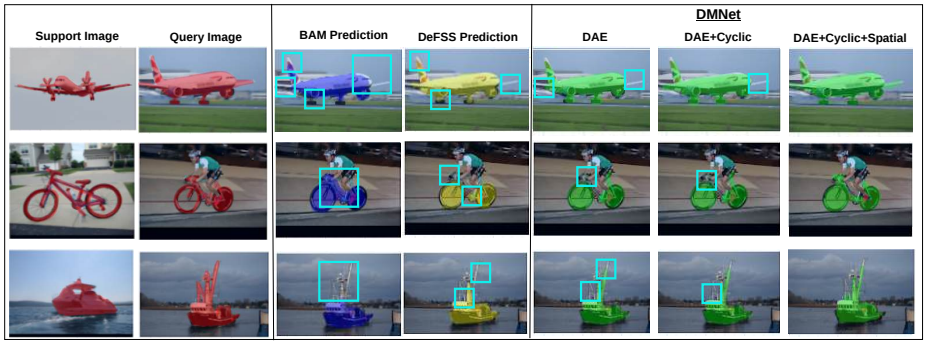


Fig. 6: Predictions of BAM [8], DeFSS [19] (Previous SOTA), DMNet (Ours). Red masks denote the ground truth, and cyan boxes denote where previous methods fail.

against representative prototype-based, attention-based, and recent correspondence-driven FSS methods under both VGG-16 and ResNet-50 backbones. Across all settings, DMNet consistently performs better than previous works on most splits. The gains are particularly notable under the more challenging COCO-20^i benchmark, suggesting that the proposed disentanglement and refinement strategies effectively enhance robustness in complex scenarios.

Table 2 further compares DMNet with foundation model-based FSS approaches. GF-SAM [34] uses DINOv2 for feature extraction and SAM as a mask generator and builds a graph structure over support-query correspondences. SANSA [3] exploits semantic priors from SAM2 to guide mask selection and refinement, while FCP [16] integrates SAM-generated masks with foreground-covering prototype matching. These approaches rely heavily on large pretrained foundation models and benefit from strong global priors, but at the cost of heavy computation. Notably, DMNet (using ResNet-50 backbone) performs strongly even without these large backbones. DMNet-SAM further benefits by replacing the frozen CNN backbone with a frozen SAM encoder without SAM prompting.

Qualitative Analysis. Figure 6 shows the prediction masks for three different methods: BAM [8] (uses a scene-based ensemble technique), DeFSS (the previous state-of-the-art method) and DMNet, using the same ResNet-50 backbone. Overall, DMNet consistently produces masks that are both more complete and more semantically aligned with the target object compared to BAM and DeFSS. While prior approaches frequently miss structural regions or become distracted by visually similar background elements (cyan boxes), especially in cases where the pose is misaligned, DMNet leverages meaningful disentanglement and robust

Table 3: Computational Efficiency and Accuracy (1-shot).

Method	Parameter Count	Inference Time (ms)	Avg mIoU
SANSA [3]	234M	67	62.3
GF-SAM [34]	180M	1800	70.1
Ours (DAE)	**5.3M**	**14.3**	**68.0**
Ours (DAE+Cyclic)	**5.3M**	**15.9**	**69.2**
Ours (DAE+Cyclic+Spatial)	**5.3M**	**19.7**	**70.8**
BAM [8]	4.9M	7.4	57.0
HMNet [31]	32.8M	118	67.8

Table 4: Ablations (ResNet-50, PASCAL-5^i).

(a) Component-wise ablation (5-shot).

Row	DAE	Cyclic	Spatial	mIoU (%)
(a)	×	×	×	62.2
(b)	✓	×	×	75.0
(c)	✓	✓	×	76.8
(d)	✓	×	✓	76.6
(e)	✓	✓	✓	**77.5**

(b) Loss-wise ablation (1-shot).

Row	L_{seg}	L_{rec}	L_{sem}	L_{ec}	L_{geo}	mIoU (%)
(a)	✓	×	×	×	×	68.9
(b)	✓	✓	×	×	×	69.8
(c)	✓	✓	✓	×	×	71.7
(e)	✓	✓	✓	✓	×	74.7
(g)	✓	✓	✓	✓	✓	**76.1**

refinement to yield better predictions with cleaner boundaries and fewer false activations, demonstrating improved support-query correspondence and generalization under large appearance variations.

Computation Cost Table 3 compares parameter count, inference latency, and FSS accuracy. Foundation-model-based approaches [3, 34] achieve competitive average performance by leveraging large pretrained backbones and promptable SAM priors. While these models benefit from large-scale pretraining and strong general-purpose representations, they incur substantial computational and memory costs that may limit practical deployment.

In contrast, classical FSS architectures such as [8, 31] are lightweight and efficient, but their accuracy lags behind DMNet's, which strikes a favorable balance between these two extremes. With only 5.3M parameters, DMNet offers a good accuracy-efficiency trade-off, with inference time comparable to lightweight baselines. The addition of the refinement modules further improves performance while maintaining low computational overhead.

6 Ablation Studies

Component-wise Ablations. Table 4 (a) shows quantitative ablation regarding the effectiveness of the components involved. Row (a) shows the baseline where we feed F_Q^{obj} to D_{seg} without any feature refinement. DAE provides the most significant performance improvement by explicitly reducing sensitivity to instance-specific variations (Row (b)), forming the foundation of the overall framework. Building on this, Cyclic Ensemble Refinement further improves performance by leveraging query-support mutual agreement (Row (c)). Similarly, Spatially Adaptive Fusion contributes complementary gains by emphasizing reliable regions while down-weighting misleading ones (Row (d)). When all three components are combined, they yield the strongest performance (Row (e)).

Figure 6 (Right) shows the qualitative ablation of DMNet's three components. The segmentation maps are refined with each component.

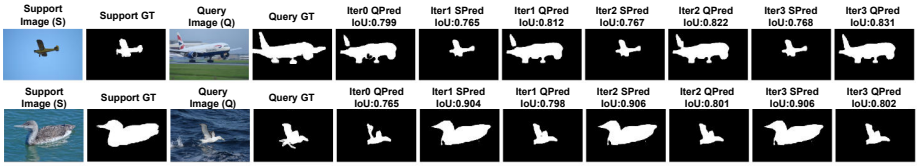


Fig. 7: Effect of Cyclic Ensemble Refinement.

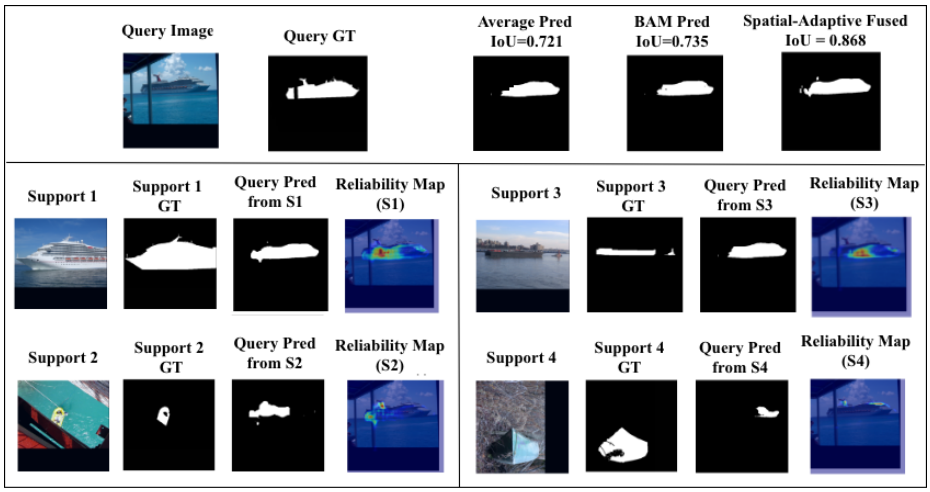


Fig. 8: Spatially Adaptive Fusion.

Effect of Cyclic Ensemble Refinement. Figure 7 shows the effect of cyclic ensemble refinement across multiple iterations. Improvements concentrate on boundaries and missing parts, and the process typically converges within two iterations, indicating a stable mutual relationship between supports and query predictions across iterations. More examples are in Supplementary.

Effect of Spatially Adaptive Fusion Module. Figure 8 contrasts different K-shot aggregation techniques - Average Fusion and BAM - with our Spatially Adaptive Fusion mechanism and visualizes the per-support reliability maps. Each reliability map shows certain warm regions indicating portions where the corresponding support is both influential and confident. Our spatial fusion mechanism automatically up-weights high-reliability areas (on the relevant object parts) and down-weights distractors (water, poles, sky in the background). This ablation shows that our Spatially Adaptive Fusion is helpful: it retains complementary evidence from helpful supports while neutralizing misleading ones, especially under appearance mismatch. More examples are included in the Supplementary.

Effect of Different Losses Table 4 (b) shows the effectiveness of different loss components involved. Incorporating L_{rec} stabilizes the decoded feature space (Row (b)), encouraging structurally coherent object representations (e.g., Fig-

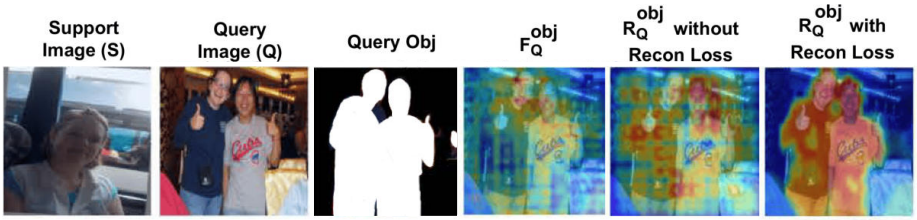


Fig. 9: Effect of $\mathcal{L}_{\text{rec}}$.

ure 9). L_{sem} further strengthens alignment between support and query features, reinforcing class-level invariance (Row (c)). Introducing L_{ec} on top of that significantly enhances geometric robustness by enforcing consistent transformation behavior, making it a core contributor to the framework (Row (e)). Finally, L_{geo} further improves the segmentation performance as it helps encode better features responsible for effective modulation by predicting the transformation parameters, resulting in more structured and interpretable modulation (Row (g)).

7 Conclusion

We present DMNet, a lightweight and effective framework for few-shot segmentation to improve robustness under pose and scale variation. Instead of explicitly aligning support and query features, DMNet disentangles semantic identity from geometric variation at the representation level. By doing this and enforcing equivariant reconstruction in a structured modulation space, the model learns object representations that are less sensitive to pose and scale discrepancies. The disentangled autoencoder refines object features before final prediction, suppressing geometry-induced distortions while preserving semantic consistency and instance-specific spatial structure. This representation-level regularization mitigates pose-scale bias without requiring explicit cross-image registration at inference. In addition, a lightweight refinement module enhances supervision from scarce supports through cyclic ensemble over augmented support views and a spatially adaptive fusion mechanism that weights pixel-level predictions according to reliability. Across standard FSS benchmarks, DMNet consistently improves segmentation accuracy, especially under large pose-scale discrepancies.

Acknowledgements

The authors were supported by NSF grants IIS-2331769, CMMI-2246673, and ECCS-2025929.

References

1. Catalano, N., Matteucci, M.: Few shot semantic segmentation: a review of methodologies, benchmarks, and open challenges. arXiv preprint arXiv:2304.05832 (2023)

2. Cohen, T.S., Welling, M.: Transformation properties of learned visual representations (2015), https://arxiv.org/abs/1412.7659
3. Cuttano, C., Trivigno, G., Averta, G., Masone, C.: Sansa: Unleashing the hidden semantics in sam2 for few-shot segmentation (2025), https://arxiv.org/abs/2505.21795
4. Dong, N., Xing, E.P.: Few-shot semantic segmentation with prototype learning. In: British Machine Vision Conference (2018), https://api.semanticscholar.org/CorpusID:52284769
5. Fan, Q., Pei, W., Tai, Y.W., Tang, C.K.: Self-support few-shot semantic segmentation. In: European Conference on Computer Vision. pp. 701–719. Springer (2022)
6. Fu, O., He, H., Li, K., Zhang, X., Zhu, L., Zeng, S., Xie, Z., Lu, Y.: Inter-and intra-image refinement for few shot segmentation. arXiv preprint arXiv:2507.05838 (2025)
7. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks (2016), https://arxiv.org/abs/1506.02025
8. Lang, C., Cheng, G., Tu, B., Han, J.: Learning what not to segment: A new perspective on few-shot segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8057–8067 (2022)
9. Lenc, K., Vedaldi, A.: Understanding image representations by measuring their equivariance and equivalence (2015), https://arxiv.org/abs/1411.5908
10. Liu, B., Jiao, J., Ye, Q.: Harmonic feature activation for few-shot semantic segmentation. IEEE Transactions on Image Processing **30**, 3142–3153 (2021). https://doi.org/10.1109/TIP.2021.3058512
11. Liu, Y., Zhu, M., Li, H., Chen, H., Wang, X., Shen, C.: Matcher: Segment anything with one shot using all-purpose feature matching. arXiv preprint arXiv:2305.13310 (2023)
12. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. CoRR **abs/1711.05101** (2017), http://arxiv.org/abs/1711.05101
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
14. Min, J., Kang, D., Cho, M.: Hypercorrelation squeeze for few-shot segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6941–6952 (2021)
15. Nguyen, K., Todorovic, S.: Feature weighting and boosting for few-shot segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 622–631 (2019)
16. Park, S., Lee, S., Seong, H.S., Yoo, J., Heo, J.P.: Foreground-covering prototype generation and matching for sam-aided few-shot segmentation. arXiv preprint arXiv:2501.00752 (2025)
17. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. CoRR **abs/1912.01703** (2019), http://arxiv.org/abs/1912.01703
18. Peng, B., Tian, Z., Wu, X., Wang, C., Liu, S., Su, J., Jia, J.: Hierarchical dense correlation distillation for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23641–23651 (2023)
19. Qin, Z., Xu, J., Ge, W.: Defss: Image-to-mask denoising learning for few-shot segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22232–22240 (October 2025)

20. Rakelly, K., Shelhamer, E., Darrell, T., Efros, A.A., Levine, S.: Conditional networks for few-shot semantic segmentation. In: International Conference on Learning Representations (2018), https://api.semanticscholar.org/CorpusID:69706665
21. Rakelly, K., Shelhamer, E., Darrell, T., Efros, A.A., Levine, S.: Few-shot segmentation propagation with guided networks. CoRR **abs/1806.07373** (2018), http://arxiv.org/abs/1806.07373
22. Saha, O., Cheng, Z., Maji, S.: GANORCON: are generative models useful for few-shot segmentation? CoRR **abs/2112.00854** (2021), https://arxiv.org/abs/2112.00854
23. Shaban, A., Bansal, S., Liu, Z., Essa, I., Boots, B.: One-shot learning for semantic segmentation. CoRR **abs/1709.03410** (2017), http://arxiv.org/abs/1709.03410
24. Shi, X., Wei, D., Zhang, Y., Lu, D., Ning, M., Chen, J., Ma, K., Zheng, Y.: Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In: European Conference on Computer Vision. pp. 151–168. Springer (2022)
25. Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., Jia, J.: Prior guided feature enrichment network for few-shot segmentation. CoRR **abs/2008.01449** (2020), https://arxiv.org/abs/2008.01449
26. Wang, H., Yang, Y., Cao, X., Zhen, X., Snoek, C., Shao, L.: Variational prototype inference for few-shot semantic segmentation. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 525–534 (2021). https://doi.org/10.1109/WACV48630.2021.00057
27. Wang, H., Zhang, X., Hu, Y., Yang, Y., Cao, X., Zhen, X.: Few-shot semantic segmentation with democratic attention networks. In: Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII. p. 730–746. Springer-Verlag, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-58601-0_43, https://doi.org/10.1007/978-3-030-58601-0_43
28. Wang, J., Zhang, B., Pang, J., Chen, H., Liu, W.: Rethinking prior information generation with clip for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3941–3951 (2024)
29. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: Panet: Few-shot image semantic segmentation with prototype alignment. In: proceedings of the IEEE/CVF international conference on computer vision. pp. 9197–9206 (2019)
30. Xu, Q., Lin, G., Loy, C.C., Long, C., Li, Z., Zhao, R.: Eliminating feature ambiguity for few-shot segmentation (2024), https://arxiv.org/abs/2407.09842
31. Xu, Q., Liu, X., Zhu, L., Lin, G., Long, C., Li, Z., Zhao, R.: Hybrid mamba for few-shot segmentation. Advances in Neural Information Processing Systems **37**, 73858–73883 (2024)
32. Xu, Q., Zhao, W., Lin, G., Long, C.: Self-calibrated cross attention network for few-shot segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 655–665 (2023)
33. Yang, B., Liu, C., Li, B., Jiao, J., Ye, Q.: Prototype mixture models for few-shot semantic segmentation. CoRR **abs/2008.03898** (2020), https://arxiv.org/abs/2008.03898
34. Zhang, A., Gao, G., Jiao, J., Liu, C., Wei, Y.: Bridge the points: Graph-based few-shot segment anything semantically. Advances in Neural Information Processing Systems **37**, 33232–33261 (2024)

35. Zhang, C., Lin, G., Liu, F., Guo, J., Wu, Q., Yao, R.: Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9586–9594 (2019). https://doi.org/10.1109/ICCV.2019.00968
36. Zhang, C., Lin, G., Liu, F., Yao, R., Shen, C.: Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. CoRR **abs/1903.02351** (2019), http://arxiv.org/abs/1903.02351
37. Zhang, G., Kang, G., Wei, Y., Yang, Y.: Few-shot segmentation via cycle-consistent transformer. CoRR **abs/2106.02320** (2021), https://arxiv.org/abs/2106.02320
38. Zhang, G., Kang, G., Yang, Y., Wei, Y.: Few-shot segmentation via cycle-consistent transformer. Advances in Neural Information Processing Systems **34**, 21984–21996 (2021)
39. Zhang, X., Wei, Y., Yang, Y., Huang, T.S.: Sg-one: Similarity guidance network for one-shot semantic segmentation. CoRR **abs/1810.09091** (2018), http://arxiv.org/abs/1810.09091
40. Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J., Barriuso, A., Torralba, A., Fidler, S.: Datasetgan: Efficient labeled data factory with minimal human effort. CoRR **abs/2104.06490** (2021), https://arxiv.org/abs/2104.06490
41. Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: Towards adapting clip for zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11175–11185 (2023)
42. Zhu, L., Chen, T., Ji, D., Ye, J., Liu, J.: Llafs: When large language models meet few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3065–3075 (2024)