DOTGraph: CLIP-Driven Feature Disentanglement and Optimal Transport based Graph Learning for Few-Shot Segmentation

Shreya Biswas and Zhaozheng Yin

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

DOTGraph: CLIP-Driven Feature Disentanglement and Optimal Transport based Graph Learning for Few-Shot Segmentation

Supplementary Material

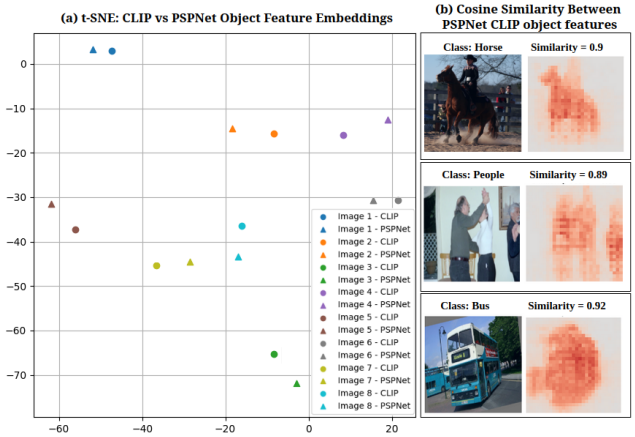

Figure 1. (a) t-SNE and (b) Cosine similarity between PSPNet and CLIP features.

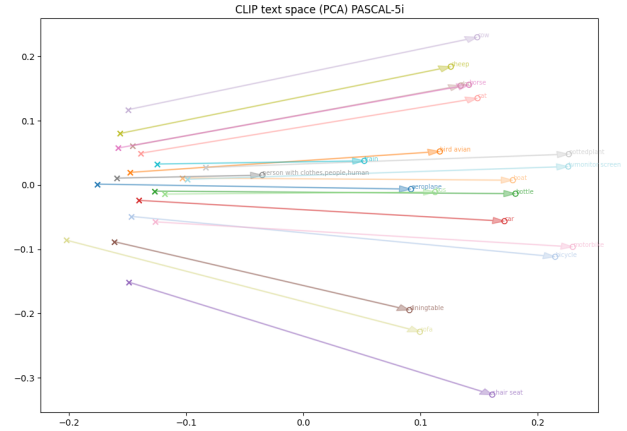

Figure 2. PCA of CLIP Foreground-Background text feature embeddings.

1. Discussions

1.1. Discussion on CLIP training space

We wanted to ensure the semantic gap between CLIP and the pretrained image encoder (PSPNet) is small and does not inherently degrade performance when the features of the two encoders are fused together. To empirically evaluate the gap, we visualize the PSPNet and CLIP object embeddings using t-SNE (Fig 1a) and report their cosine similarity scores (Fig 1b). The two feature spaces are clustered across diverse classes and the consistently high cosine similarity scores (≥ 0.89) reinforce that despite different training domains, both encoders produce highly aligned semantic embeddings for object regions. Our Feature Refinement Stage further mitigates the residual semantic gap (if any).

Analysis of CLIP Text Embedding Space We analyze the CLIP text embeddings corresponding to the foreground (FG) prompts and background (BG) prompts using a 2D PCA projection (Figure 2). The visualization reveals the following key observations:

1. *Separation of FG and BG embeddings:* Across almost all classes, the FG embeddings (circles) are consistently shifted away from their corresponding BG embeddings (crosses). This demonstrates that CLIP distinguishes between "presence" and "absence" prompts, even though both share the same class name.
2. *Directionality of FG–BG offsets:* The arrows representing BG $\rightarrow$ FG transitions show a consistent left-to-right orientation in the embedding space. This suggests that CLIP encodes "absence vs. presence" as a systematic and semantically meaningful direction.
3. *Class-specific variations:* For certain categories (e.g., *sofa/diningtable* or *car/motorbike/bicycle*), the magnitude of the BG $\rightarrow$ FG offset varies. Longer offsets indicate that CLIP makes a stronger distinction between FG and BG wording, whereas shorter offsets imply weaker separability between the two.
4. *Cluster structure:* Foreground embeddings cluster according to semantic similarity (e.g., *cow, sheep, horse, cat* form a group, and *bicycle, motorbike* are closely aligned). By contrast, background embeddings are more compact and show less semantic variation, consistent with the generic nature of "a photo without {class}" prompts.

Implications for our method: This analysis confirms that BG text prompts are not randomly distributed in CLIP space, but rather form a structured and semantically distinct region from FG prompts. The BG $\rightarrow$ FG offset can therefore be exploited: our feature disentanglement module leverages this systematic shift to suppress background bias and enhance foreground-object awareness in few-shot segmentation.

1.2. Discussion on CLIP-based Background features

The background text prompt feature indeed can be anywhere in the CLIP embedding space. To relate it with the support image provided, we extract the support image fea-

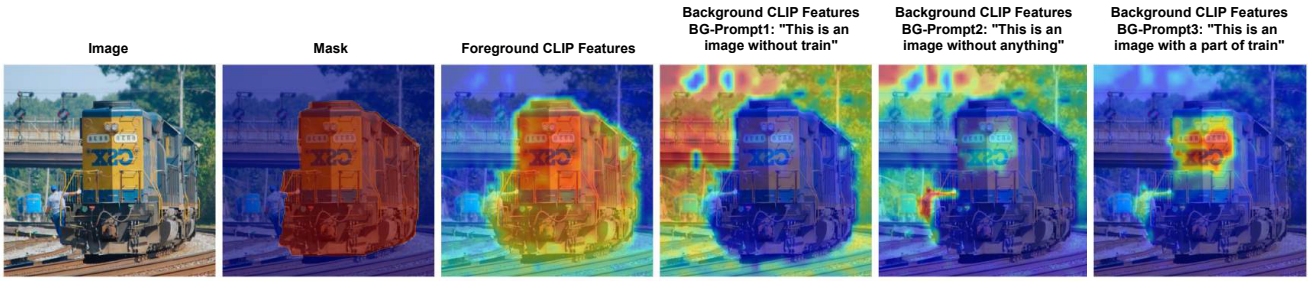

Figure 3. Effect of different background prompts.

tures (from CLIP's visual encoder forming F_q) to get the features of the support background. A global average pooling operation is then applied across spatial positions, yielding the aggregated query token:

$$v_q = \frac{1}{hw} \sum_{i=1}^{hw} F_q(i).$$

Next, we compute similarity scores between the pooled query token and each text embedding. To sharpen these scores, we apply a softmax normalization with temperature

$$S_i = \text{softmax}\left(\frac{v_q^\top F_i^t}{\|v_q\| \, \|F_i^t\| \cdot \tau} \right), \quad i \in \{f, b\},$$

where f and b denote the foreground and background, respectively.

To localize discriminative regions in the image, we backpropagate the classification score and calculate gradient-based weights for each feature map channel:

$$w_m = \frac{1}{hw} \sum_{i,j} \frac{\partial S_f}{\partial F_q^m(i,j)},$$

where $F_q^m(i,j)$ is the activation of the m-th channel at spatial position (i,j). These weights emphasize the channels most responsible for aligning the query with the foreground text embedding. Finally, we generate the visual–text prior heatmap by combining the weighted feature maps and applying a ReLU activation:

$$P_{vt} = \text{ReLU}\left(\sum_m w_m F_q^m \right), \quad P_{vt} \in \mathbb{R}^{1 \times h \times w}.$$

This heatmap serves as a Grad-CAM-style prior that highlights regions corresponding to the textual prompts, and in particular, allows us to extract background-aware features using the background prompts.

We provide some additional experiments we ran on the background prompts and how the changes affected our FSS results:

Table 1. Effect of different background (BG) text feature choices on PASCAL-5i 5-way-5-shot FSS performance.

Dataset	BG Text Features	mIoU	FB-IoU
PASCAL	No BG Prompt	74.7	84.34
PASCAL	BG-Prompt1 (Ours)	77.5	87.57
PASCAL	BG-Prompt2 (Class Agnostic)	75.2	85.46
PASCAL	BG-Prompt3	72.2	82.19

1. No Background Prompt at all: Essentially, no Feature disentanglement stage.
2. Background Prompt 1 (BG-Prompt1): "This is an image without <class label >"
3. Background Prompt 2 (BG-Prompt2): "This is an image without anything"
4. Background Prompt 3 (BG-Prompt3): "This is an image with part of a <class label >"

This shows that our Background prompt feature is essential to the FSS performance.

1.3. Optimal Transport Intuition.

In our setup, the support and query features are treated as two discrete distributions over spatial regions. Each support patch feature v_j^S carries a unit of probability mass, and each query patch feature v_i^Q is a site to which this mass can be transported. Optimal Transport (OT) computes the minimal-cost transport plan that redistributes support mass across query locations while satisfying global consistency constraints. This differs from standard attention, which only computes local similarity scores independently for each query node.

Formulation. We model the support and query feature sets as discrete probability measures:

$$\mu = \frac{1}{N} \sum_{j=1}^{N} \delta_{v_j^S}, \qquad \nu = \frac{1}{N} \sum_{i=1}^{N} \delta_{v_i^Q}, \tag{1}$$

where δ_x is the Dirac distribution at feature x, and $N = H \times W$ is the number of spatial nodes.

The cost of aligning support node j to query node i is given by the squared Euclidean distance in embedding space:

$$C_{ij} = \|v_i^Q - v_j^S\|_2^2. \tag{2}$$

OT seeks a coupling $T \in \mathbb{R}^{N\times N}$ with marginals μ, ν:

$$T\mathbf{1} = \mu, \qquad T^\top \mathbf{1} = \nu, \tag{3}$$

that minimizes the total transport cost:

$$\min_T \langle T, C\rangle = \sum_{i,j} T_{ij} C_{ij}. \tag{4}$$

To make this tractable, we adopt entropic regularization, yielding the Sinkhorn objective:

$$\min_T \langle T, C\rangle - \varepsilon H(T), \qquad H(T) = -\sum_{i,j} T_{ij} \log T_{ij}, \tag{5}$$

where ε is the regularization strength. The solution is obtained efficiently by iteratively scaling rows and columns of the Gibbs kernel $K = \exp(-C/\varepsilon)$.

Comparison to Attention. Standard attention computes

$$A_{ij} = \text{softmax}_j(v_i^Q \cdot v_j^S), \tag{6}$$

so each query node i independently attends to all support nodes. This independence often produces diffuse maps where background regions receive non-negligible weights. In contrast, OT solves

$$T^* = \arg\min_T \langle T, C\rangle - \varepsilon H(T), \quad T\mathbf{1} = \mu,\ T^\top \mathbf{1} = \nu, \tag{7}$$

which enforces a globally consistent transport plan. As a result, query background nodes cannot all receive high weights simultaneously, since the marginal constraints prevent mass overflow. This yields sharper and more semantically faithful correspondences (see Fig. 5), with query-object regions consistently aligned to support-object regions while suppressing object-background matches.

1.4. Adaptability to multiple classes

DOTGraph is modular and can be parallelized across-classes, similar to prototype-based methods. Feature extraction is shared across classes. Class-specific graph refinement is performed on low-resolution representations and can be efficiently parallelized using batched operations. As the number of classes increase, the computational overhead can be significantly reduced with GPU parallelism.

Figure 4. Feature activations with L_{GCL} and L_{GCL2}.

Dataset	S1	S2	S3	S4	Mean
DOTGraph with L_{GCL}	77.5	82.3	74.9	73.8	77.1
DOTGraph with L_{GCL2}	77.9	82.0	75.1	73.8	77.2

Table 2. 5-way-1-shot mIoU (PASCAL-5^i, ResNet50).

1.5. Discussion on effects of query background:

We mitigate the effect of the query background Q_{bg} through *Contrastive Graph Supervision* (Eq. 9 of the Main Paper), aligning foreground features while pushing them away from support background; and *CLIP-Driven Background Filtering*, highlighting object regions and suppressing background in both Support S and Query Q (Fig. 5 of the Main Paper). To further check the effectiveness of our formulation, we experimented further by modifying Eq. 9 as L_{GCL2}:

$$\mathcal{L}_{GCL2} = -\frac{1}{N}\sum_i \log \frac{\exp\left(\frac{\text{sim}(\mathbf{G}_Q^{obj,i}, \mathbf{G}_S^{obj,i})}{\tau}\right)}{\exp\left(\frac{\text{sim}(\mathbf{G}_Q^{obj,i}, \mathbf{G}_S^{obj,i})}{\tau}\right) + \exp\left(\frac{\text{sim}(\mathbf{G}_Q^{obj,i}, \mathbf{T}_S^{bg,i})}{\tau}\right) + \exp\left(\frac{\text{sim}(\mathbf{G}_Q^{obj,i}, \mathbf{G}_Q^{bg,i})}{\tau}\right)}$$

L_{GCL2} leads to a very small improvement (Table 2), suggesting that L_{GCL} mitigates background noise on par with L_{GCL2}. Fig 4 shows that the activations are only a bit more focused on the object regions with L_{GCL2}. Thus, our formulation of Graph Contrative Learning Loss L_{GCL} is robust and is effective in removing the unwanted effect of the background without additional factors.

1.6. Relation between Base and Novel Classes

We wanted to check if relation between the base and novel classes have any impact on DotGraph's performance. We note that the datasets do not contain examples where novel classes which are subparts of base classes. The gap between base and novel classes is often not a traditional domain gap but a semantic granularity gap. DOTGraph correctly identifies novel classes, unlike prototype-based methods (Fig 5).

1.7. Pose-Scale Discrepancy and how OT-based graph learning helps solve it

Pose-scale discrepancy means the **query and support features do not align one-to-one**:

- Small support object vs. large query object → the support prototype covers only a fraction of the query space.
- Different poses/orientations → features appear dissimilar even if semantically they represent the same object part.
- Some object parts may be absent in either support or query.

Cross-attention fails here because it is *local, similarity-driven*: it only connects pairs of features that already look alike. If no direct match exists, it produces weak or missing correspondences.

Why OT-based Graph Learning Helps.

- *Global distribution alignment (not pairwise only):* OT does not require each query pixel to find its single closest support pixel. Instead, it computes a *transport plan* that redistributes the **entire support distribution** over the query distribution. This allows the model to capture correspondences even when features are not locally aligned.
- *Mass-preserving mapping:* In OT, the "mass" of support features must be spread over the query features.
  - If the query object is larger, OT ensures the limited support mass is distributed across all query regions, mitigating scale differences.
  - If some parts are missing in either support or query, OT redistributes mass to the remaining regions, preventing collapse.
- *Integration with graph learning:* By formulating features as graphs (nodes = spatial regions, edges = similarities), the OT transport plan becomes a *message-passing mechanism*: it globally shifts query-object node embeddings closer to the support-object distribution while pushing them away from support background. This enforces alignment at the *distribution level*, not just local dot-product similarity.

1.8. Support-Query Attention and how it helps.

Previous cross-attention methods in few-shot segmentation, such as CyCTR [9] and SCCAN [7], have shown that pixel- or patch-level matching between support and query can improve segmentation performance. However, they face two limitations that prevent them from fully capturing fine object details:

- **Noisy correspondences:** Similarity-based attention often aggregates irrelevant support pixels into query regions, which blurs boundaries and loses subtle details. CyCTR mitigates this by cycle-consistency filtering, but still relies on nearest-neighbor correspondences, which may suppress thin or rare object parts when they are weakly matched.
- **Foreground–background entanglement:** As noted in SCCAN, query background features can be mistakenly fused with support foreground, leading to inaccurate edge localization and loss of high-frequency details.

Our proposed method is explicitly designed to preserve fine object structures:

- **Dual-branch design:** We combine OT-based graph learning for global distribution alignment with SQA for fine-scale refinement. This ensures that the query is semantically aligned before attention focuses on detail regions.
- **Detail-preserving fusion:** Unlike CyCTR and SCCAN, which mainly act as noise suppression mechanisms, SQA explicitly enhances boundary- and part-level correspondences by leveraging higher-resolution features. This enables sharper segmentation masks and better preservation of thin structures.
- **Complementarity:** While prior works emphasize avoiding harmful correspondences, our SQA functions as a *refinement stage* after disentanglement and OT alignment, selectively amplifying fine-grained object signals rather than compressing or filtering them away.

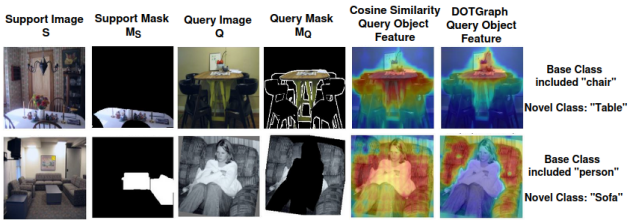

Figure 5. DOTGraph's activation on novel classes.

2. More implementation details

As in standard FSS pipelines, images are resized to a fixed resolution (225×225). Below are information on more implementation details of our architecture.

2.1. Decoder

For a fair comparison, we use the ensemble module following [1] to filter the categories appearing in the training process for the query mask prediction, We also feed the refined support features from the graph into the decoder to predict the support object mask for additional supervision. PSPNet[10] serves as the base learner for our experiments.

2.2. Feature Extraction module

To enhance feature resolution, we adopt the PPM module[10], a widely used component in semantic segmentation models. This module refines multi-scale contextual features after the 4th block of the backbone, generating a prior mask following [5]. Finally, the predicted segmentation maps are resized to match the original input dimensions, ensuring accurate comparison with the ground-truth labels.

2.3. Formulation of the Query Object Feature F_{obj}^{Q}

We concatenate features from layer2 and layer3 (mid-level and high-level semantics) from pretrained PSPNet to serve as the multi-scale visual representations for both query and support images, retaining both semantic richness and spatial detail. Then, we use the resized masks for a Weighted Masked Average Pool operation to generate the features of

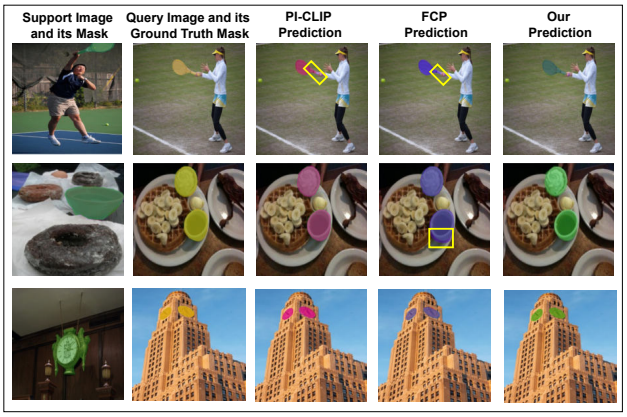

Figure 6. Predictions on COCO-20i dataset.

the object in both the support and query images - forming F^S_{obj} and F^Q_{obj} respectively.

3. Additional Visualizations

3.1. Segmentation Map Prediction on COCO-20i Dataset

Figure 6 shows the segmentation map prediction by recent state-of-the-art works - PI-CLIP [6] and FCP [3], and our proposed method DOTGraph. We see DOTGraph is able to capture finer details when the other two methods cannot (as seen by the yellow boxes - which highlight the limitations of the state-of-the-art methods). Becausew COCO-20i is a challenging dataset than PASCAL-5i, even minute differences are non-trivial and of vital importance for accurate segmentation.

3.2. Feature Visualizations

Figure 7 shows additional examples of the effect of our framework on the features extracted for the object for Q and S. The dual-branched graph can visibly improve the focus on the query object feature, and reduce the dependency on the support background features (for context) for mask prediction.

4. Comparision with Foundation Model-based Architectures

Since DOTGraph incorporates CLIP as an additional feature extractor, we compare our performance with other emthods using Foundation models like CLIP, DINOv2 or SAM as feature encoders. We do not include these results in the main DOTGraph's. Table 3 shows the results. We notice that even when compared with models using powerful feature extractors like DINOv2 or SAM, DOTGraph outperforms all of them.

5. Computational Analysis of DOTGraph

Our dual-branch graph update module introduces two sources of computational overhead: (i) the Optimal Transport (OT) branch, and (ii) the Support-Query Attention (SQA) branch. Let the downsampled feature map have spatial size $H \times W$, so that the number of nodes is $N = H \cdot W$, and let C denote the channel dimension of node features. The projection dimension in the attention branch is d, and the number of Sinkhorn iterations used in OT is S.

Optimal Transport branch. Solving the exact OT has a time complexity of $\mathcal{O}(N^3)$. But our goal is to reduce the complexity by approximating the OT correspondence using Sinkhorn iterations. For each query-support pair we construct two cost matrices (object vs. object, object vs. background):

$$C^{\text{obj}}_{ij} = \|v^{(Q)}_i - v^{(S)}_j\|^2_2, \quad C^{\text{bg}}_{ij} = \|v^{(Q)}_i - v^{(S,bg)}_j\|^2_2,$$

each of size $N \times N$. Computing these costs requires $\mathcal{O}(N^2C)$ operations, and storing them requires $\mathcal{O}(N^2)$ memory. Sinkhorn iteration scales as $\mathcal{O}(SN^2)$, so the total cost per OT branch is

$$\mathcal{O}(N^2(C+S)), \quad \text{space } \mathcal{O}(N^2).$$

Support-Query Attention branch. Scaled dot-product attention requires projections $Q = VW_Q$, $K = VW_K$, $V = VW_V$, costing $\mathcal{O}(NCd)$. Computing the score matrix $QK^\top$ and message AV both require $\mathcal{O}(N^2d)$. Thus the attention branch costs

$$\mathcal{O}(NCd + N^2d), \quad \text{space } \mathcal{O}(N^2).$$

Overall complexity. Each graph iteration therefore requires

$$\boxed{\mathcal{O}\big(N^2(C+S+d) + NCd\big) \text{ time}, \quad \mathcal{O}(N^2) \text{ space}}$$

with T graph iterations yielding a factor T. The quadratic N^2 term dominates, but N is small because we operate on downsampled feature maps (e.g. 60×60 resolution, $N = 3600$). In this case, a dense $N \times N$ matrix occupies $\sim$50 MB in float32, so storing multiple such maps (OT cost, transport plan, and attention map) remains tractable on modern GPUs.

Practical efficiency. With $C = 256$, $d = 128$, $S = 10$, $N = 3600$, and $T = 2$, the dual-branch module requires on the order of 9–10 G multiply–adds per query–support pair, which is negligible compared to the frozen backbone forward pass. Runtime measurements confirm that the graph refinement adds only tens of milliseconds per

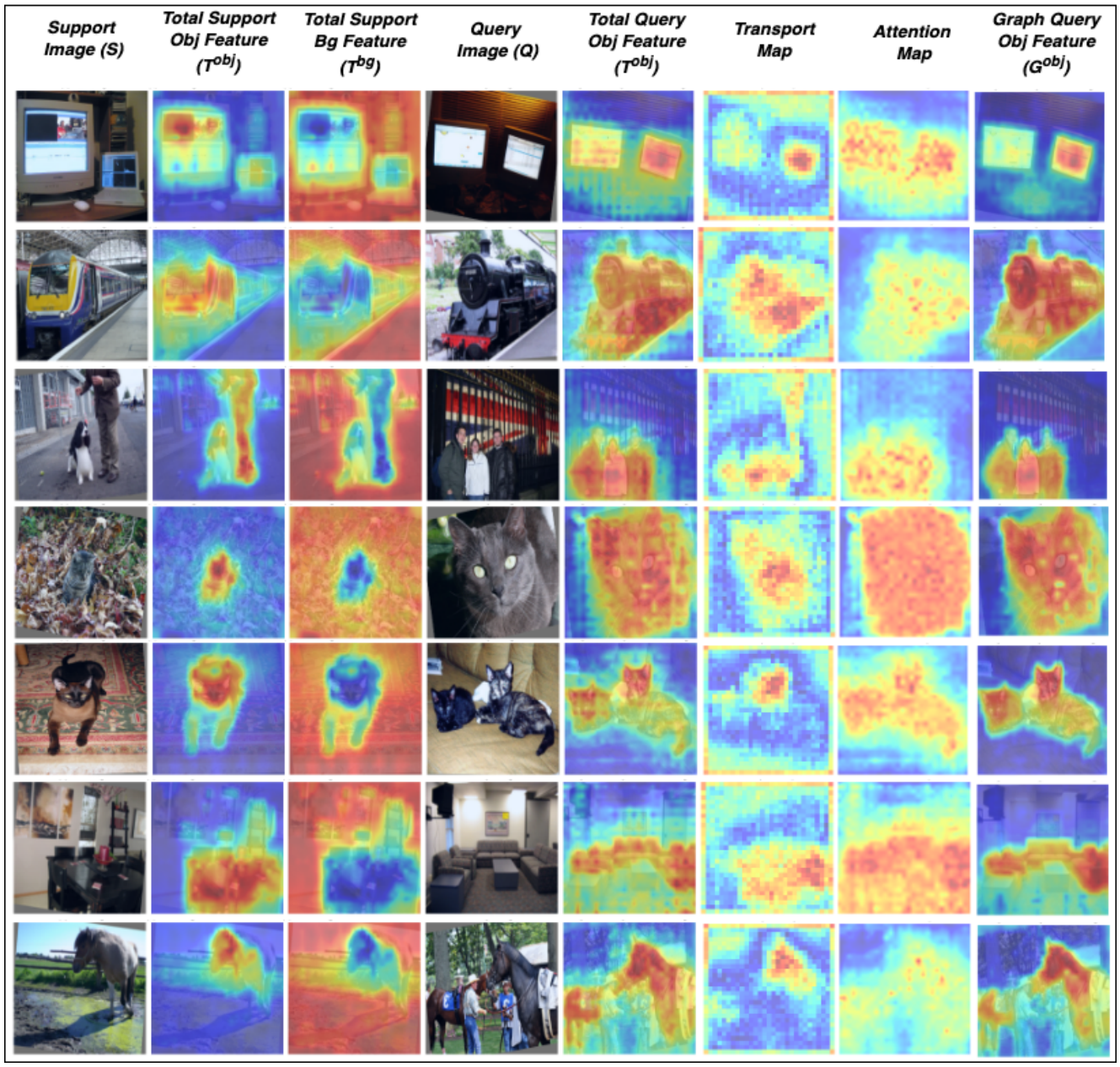

Figure 7. Feature Heatmaps showing the effect of our GNN.

Table 3. Comparing experimental results on the PASCAL-5^i and COCO-20^i with Foundation models-based FSS architectures. F means the type of image feature encoder. S1-S4 denote the splits. mIoU is the evaluation metric. Bold: best, underline: second best.

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
PI-CLIP + HDMNet [6]	ResNet-50	76.4	**83.5**	74.7	72.8	76.8	76.7	83.8	75.2	73.2	77.2	49.3	65.7	55.8	56.3	56.8	56.4	**66.2**	55.9	58.0	59.1
Matcher [2]	SAM+DINOv2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	52.7	-	-	-	-	52.4
GF-SAM [8]	SAM	-	-	-	-	71.9	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
VRP-SAM [4]	ResNet-50+SAM	73.9	78.3	70.6	65.0	71.9	-	-	-	-	-	48.1	55.8	60.0	51.6	53.9	-	-	-	-	-
FCP [3]	ResNet-50	74.9	77.4	71.8	68.8	73.2	77.2	78.8	72.2	67.7	74.0	46.4	56.4	55.3	51.8	52.5	52.6	63.3	**59.8**	56.1	58.0
DOTGraph (Ours)	ResNet-50	**77.5**	82.3	**74.9**	**73.8**	**77.1**	**77.9**	**84.6**	**75.7**	**74.0**	**78.1**	**49.8**	**65.8**	**56.2**	**57.4**	**57.3**	**56.6**	65.6	58.8	**58.3**	**59.8**

Table 4. Computational Cost (100 Epochs, Dataset: PASCAL-5^i, Backbone: ResNet50)

Model	Training Time (hrs)	Inference Time (min)	No. of Learnable Params
PI-CLIP [6]	11.5	17	153786055
DOTGraph	14.2	18	170998535

episode, while significantly improving accuracy. Thus, although our formulation is theoretically quadratic in N, it is lightweight in practice due to the low-resolution graphs and the small number of iterations.

Table 4 shows that the computational cost is only marginally higher than PI-CLIP. Note that DOTGraph leverages pretrained PSPNet and CLIP encoders. Only the Feature Refinement and the Segmentation stages are learnable. OT and attention pathways use shared feature volumes, with minimal overhead - OT is computed using $\leq$3 Sinkhorn iterations, and attention uses a single-head configuration.

The one-time training cost is slightly higher than PI-CLIP, but the inference time is negligible, considering the gain in performance we get.

References

[1] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. 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*, pages 8057–8067, 2022. 4

[2] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. *arXiv preprint arXiv:2305.13310*, 2023. 6

[3] Suho Park, SuBeen Lee, Hyun Seok Seong, Jaejoon Yoo, and Jae-Pil Heo. Foreground-covering prototype generation and matching for sam-aided few-shot segmentation. *arXiv preprint arXiv:2501.00752*, 2025. 5, 6

[4] 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, 2024. 6

[5] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *CoRR*, abs/2008.01449, 2020. 4

[6] Jin Wang, Bingfeng Zhang, Jian Pang, Honglong Chen, and Weifeng Liu. Rethinking prior information generation with clip for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3941–3951, 2024. 5, 6, 7

[7] Qianxiong Xu, Wenting Zhao, Guosheng Lin, and Cheng Long. Self-calibrated cross attention network for few-shot segmentation, 2023. 4

[8] Anqi Zhang, Guangyu Gao, Jianbo Jiao, Chi Harold Liu, and Yunchao Wei. Bridge the points: Graph-based few-shot segment anything semantically, 2024. 6

[9] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Advances in Neural Information Processing Systems*, 34:21984–21996, 2021. 4

[10] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016. 4