Flexible visual prompts for in context learning in computer vision

Thomas Foster **Ioana Croitoru** **Robert Dorfman** **Christoffer Edlund**
Thomas Varsavsky **Jon Almazán**
V7 Labs
{thomas,ioana,robert,christoffer,tom,jon}@v7labs.com

Abstract

In this work, we address in-context learning (ICL) for the task of image segmentation, introducing a novel approach that adapts a modern Video Object Segmentation (VOS) technique for visual in-context learning. This adaptation is inspired by the VOS method's ability to efficiently and flexibly learn objects from a few examples. Through evaluations across a range of support set sizes and on diverse segmentation datasets, our method consistently surpasses existing techniques. Notably, it excels with data containing classes not encountered during training. Additionally, we propose a technique for support set selection, which involves choosing the most relevant images to include in this set. By employing support set selection, the performance increases for all tested methods without the need for additional training or prompt tuning. The code can be found at https://github.com/v7labs/XMem_ICL/.

1 Introduction

In-context learning (ICL) is a new paradigm primarily derived from large autoregressive language models, notably GPT-3 [6]. Instead of updating model weights, ICL employs domain specific input-output pairings, termed *prompts* or a *support set*, which are integrated at test time to direct the model towards the intended output. This approach has proven effective in various applications with large language models (LLMs), including question-answering [21, 18], translation [23, 2] and sentiment analysis [3]. For example, at test time, a model can be guided to perform sentiment analysis by providing a structured input that pairs a statement with its sentiment label [3].

ICL offers the benefits of few-shot learning [20, 34] without the need for weight modification or re-training when switching tasks or domains. For enterprises deploying models, ICL is desirable, allowing economical deployment of a single model that can be used for various tasks, domains, or datasets through tailored prompts, without requiring re-training. Moreover, ICL models can adapt to outdated training data or newly available data by simply using different prompts at inference.

In-context learning is increasingly explored in linguistics, but its application in computer vision remains limited. Among the few studies exploring this area, three recent ones [5, 32, 33] are notable. They introduce a visual prompt by merging a support set of relevant image-label pairs into a large grid. However, this 'gridding' approach has several drawbacks: it is resource-intensive, imposes limits on the maximum resolution for inference, and fixes the grid size (i.e. support set size) for all future inferences once a model is trained.

In this paper, we are the first to explore the adaptation of recent Video Object Segmentation (VOS) methods to visual ICL, aiming to overcome the limitations of existing approaches. VOS methods, aimed at binary semantic segmentation, sequentially construct a memory of the video frames and masks. Given their design to handle videos, potentially encompassing thousands of frames, VOS

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023.

methods allow for the efficient utilisation of a bigger support set. Additionally, during inference, if multiple queries are to be processed using the same support set, the memory of the VOS method can be efficiently cached to avoid redundant computations. Each frame is processed on its own, ensuring that the maximum resolution of each image is not constrained by the size of the support set. Unlike existing ICL methods, the size of the support set is flexible, allowing VOS methods to adapt at inference time if more examples become available.

Furthermore, we conduct empirical evaluations to comprehensively compare existing ICL methods with VOS methods. This comparison specifically targets the task of binary segmentation, where each pixel in an image is classified as either foreground or background. This pixel-wise classification is crucial for various applications, such as medical imaging (segmenting teeth) [17, 1], disaster response (identifying flooded areas) [26], biomedical research (cell segmentation) [12], and infrastructure maintenance (segmenting potholes and cracks in roads) [30].

Our study reveals that VOS methods exhibit superior generalization to unseen classes –classes not encountered during the training process– compared to previous techniques, while also being computationally efficient. We additionally explore and demonstrate the significant influence of support set selection on the performance of visual ICL methods. We show that including examples in the support set that are similar to the target query boosts overall performance. Our final method, which adapts a VOS approach to ICL for binary segmentation and incorporates support set selection, outperforms all other ICL methods on unseen classes.

Our main contribution can be summarized as follows: (1) We are the first to study and adapt Video Object Segmentation (VOS) methods for visual in-context learning (ICL), thus removing the restrictions of existing ICL methods on support set size and image resolution. (2) We compare VOS methods with previous visual ICL methods, revealing that our proposed VOS method performs significantly better on unseen classes and datasets. (3) Through extensive experiments, we demonstrate the importance of support set selection in enhancing visual ICL performance. By leveraging semantic visual similarity, our approach yields substantial improvements across all tested ICL methods.

2 Related work

'Gridding' methods. Current in-context learning (ICL) methods [5, 32, 33] form a visual prompt by combining a support set of relevant prompts along with their masks into a large grid. Through pre-training a neural network to fill in gaps in this grid, they enabled the model to apply in-context learning to new tasks. Painter [32] pre-trains a vanilla vision Transformer (ViT) using a 2x2 grid of images, on a range of both segmentation (semantic, panoptic etc) and non-segmentation tasks (monocular depth estimation, image restoration etc). In a follow up work [33] the authors further optimise this method for segmentation tasks, by introducing a random recolouring data augmentation at training time in an attempt to force the model to derive colouring information from the context, instead of internalising specific colours. [5] also forms a 2x2 gridded prompt, but explores various architectures for the in-painting task, such as VQGAN [13], BEiT [4] and MAE [15].

Larger support sets enhance model performance by providing more information. However, once trained, these models are constrained to the support set size they were initially configured with. Adjusting to a different support set size is not inherently supported and it necessitates the application of various techniques such as feature averaging. Other solutions might involve scaling to larger grid sizes which requires substantially more compute or downscaling the resolution. Unlike these models, VOS methods can dynamically adapt to any support set size at test time.

VOS methods. XMem [9] and STCN [10] are the most recent in a long line of Video Object Segmentation (VOS) methods that process frames and associated support masks one by one, embedding them into a "memory" that is then decoded into a new mask prediction. STCN [10] was the first to simplify this process to use just two ResNet [16] networks, one for embedding frames and masks respectively. XMem extended STCN, adding "long-term" memory for object tracking in long (1000+ frame) videos.

AOT [36] is another approach to VOS that leverages transformer style attention over encodings of frames. The main contribution of AOT is predominantly how to track multiple objects at once, instead of producing binary masks like most VOS methods. In this work, we focus on the XMem family of VOS methods, since it represents the state of the art in binary segmentation of long videos.

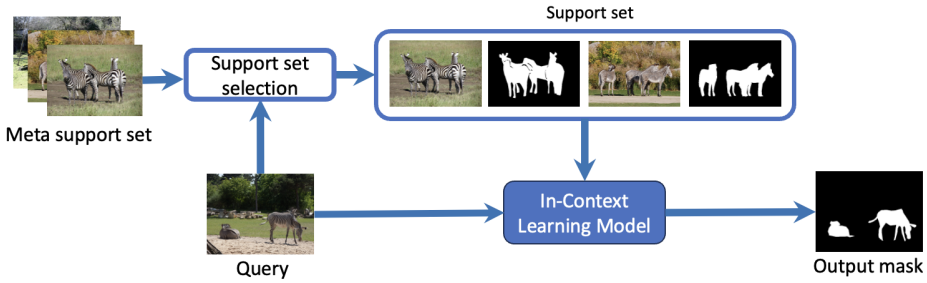

Figure 1: In-context learning for binary segmentation with support set selection. A support set is selected from a meta-support set based on similarity to the query image. This method effectively segments the query image, even in cases where it contains objects belonging to classes not seen during training.

Other methods. UniverSeg [7] is a notable alternative to the gridding approach. They focus on medical imaging and adapt a UNet [28] architecture to leverage a support set via repeated applications of a novel "CrossBlock" module that uses shared convolutions between the test image and each item in the support set. The model is trained on low resolution 128x128 medical imaging tasks. PerSAM [37] is notable in that it attempts to adapt the powerful SAM [19] model for the ICL setting. The process begins by executing SAM inference on a single image from the support set, during which the model's internal activations and embeddings are saved. These saved states are then utilized to influence a subsequent SAM run on the test image, aiming to segment the object identified in the support set.

Few-shot semantic segmentation methods. Few-shot segmentation (FSS) methods, such as [35, 24, 27, 31], traditionally fine-tune model parameters based on a provided support set. In contrast, in-context learning does not need retraining if new examples became available. A standout benefit of in-context learning is its swift adaptability to new classes or tasks, eliminating the computational burden associated with weight updates.

Support set selection. Research in the area of support set selection is limited. Some works, such as [32], iterate over the entire meta-support set in search of the prompt yielding the highest performance. Additionally, they perform experiments where the prompt is learned through backpropagation. However, these methodologies require both training phases and access to labels of the whole meta-support set. This requirement is an important distinction from the approach proposed in our study. Our method is designed to efficiently select support sets without the need for extensive training or access to additional labels, thereby offering an effective alternative to the existing strategies.

3 Method

3.1 Task Definition

While ICL often involves seamless task-switching, our paper narrowly focuses on proposing a method for the task of binary segmentation. The task of binary segmentation is defined as: given an image $I \in \mathcal{R}^{H \times W \times C}$, produce a binary mask $M \in \{0,1\}^{H \times W}$ that identifies the pixels containing the class of interest. Despite this specialization, we continue to employ the principles of in-context learning as our proposed approach demonstrates robust generalization to unseen data. For the purposes of this study, unseen data refers to classes that the model has not encountered during any training that involves weight updates.

Given a dataset $S = ((I_1, M_1), ...(I_n, M_n))$ where I_i is an image and M_i a binary mask, we divide S into a set $S_{\text{meta-support}}$ of meta-support pairs and an evaluation set S_{eval}. At inference time given an image $I_i \in S_{\text{meta-support}}$, a segmentation ICL method is a map from the meta-support set and the test image to a mask prediction: $(I_i, S_{\text{meta-support}}) \mapsto M_{\text{pred}_i}$ We will discuss how it can be beneficial to only use a subset S_{support} of $S_{\text{meta-support}}$ for each image in S_{eval}, which we call **support set selection**. The overview of the mapping $(I_i, S_{\text{meta-support}}) \mapsto M_{\text{pred}_i}$ using support set selection is visualized in Figure 1.

Figure 2: Construction of multiple binary segmentation datasets from one semantic segmentation dataset. Each image-mask pair in the semantic segmentation dataset is divided into multiple image-mask pairs for binary segmentation, one for each class present in the semantic segmentation.

3.2 Visual ICL with XMem

As mentioned before, in this study we focus on the XMem family of VOS methods. XMem [9] employs a distinctive memory architecture with three interconnected memory stores: a frequently updated sensory memory, a high-resolution working memory, and a compact long-term memory. It consists of three end-to-end trainable networks: a *query encoder* extracting features from the query image, a *decoder* generating object masks from memory output, and a *value encoder* merging image and mask to derive new memory features. Unlike 'gridding' in-context learning methods, XMem handles individual images and masks, aligning more with prevalent image-to-embedding approaches in vision research.

XMem begins by storing the initial embeddings of image-mask pairs in memory. When a new frame is introduced, XMem compares its embedding pixelwise to all embeddings stored in memory. This comparison results in an affinity map that highlights the regional similarities between frames. Subsequently, the decoder generates a new mask based on this affinity map. This newly created mask is embedded into the memory to be utilized for processing subsequent frames. To enhance this process further, XMem incorporates an additional long-term memory component. When the capacity of the original memory is reached, it undergoes condensation through the application of a k-nearest neighbour algorithm. For further details, we refer the reader to [9].

To adapt XMem for in-context learning, we initially populate the model's memory with embeddings of image-mask pairs derived from the meta-support set $S_{\text{meta-support}}$. XMem inference is then run on the test image, leveraging the memory built up from the meta-support set.

3.3 Support set selection

Despite the fact that we can use up to over a thousand pairs as a support set for a VOS method, we found that segmentation performance plateaued after support sets of around size 20. This motivated the exploration of how best to choose the support set from a large pool of labelled images to choose from. We hypothesised that having images similar to the query in the support set would help performance. To test this hypothesis, we extract image embeddings (such as CLIP [25]) from $S_{\text{meta-support}}$. At test time, for each test image I_{test}, we select the support set $S_{\text{support}} \subseteq S_{\text{meta-support}}$ as the N nearest neighbours to the embedding of I_{test} from $S_{\text{meta-support}}$. It is important to note that unlike other methods from the literature [33, 32] which perform prompt tuning, our strategy for support set selection does not demand additional training. We refer to XMem models which use support set selection as **XMem+s^3**.

3.4 Training

Given that XMem is trained in a supervised manner on video frames – with minimal appearance changes between consecutive frames – we hypothesize that training the model on a dataset comprised of diverse support set images can enhance its performance. To test this hypothesis, we produced a large binary segmentation dataset. This dataset was constructed by taking semantic segmentation datasets and splitting them up into multiple binary segmentation datasets, as presented in Figure 2. Unlike many existing few-shot segmentation datasets such as FSS1000 [22], our approach provides numerous examples per class. We are training XMem on this dataset.

The training process begins by randomly selecting a class from a training dataset and sampling 16 image-mask pairs from its meta-support set. Treating these image-mask pairs as video frames, we conduct one training step as in the original XMem training procedure. In the end we train the VOS method on the whole dataset resulting in a single XMem finetuned model. We refer to models trained on still images as **XMem+T**.

4 Experimental Results

We begin by introducing the datasets used for training and evaluation in Section 4.1. This is followed by ablation studies in Section 4.2 and finally we present a comparative analysis against other methods in Section 4.3.

4.1 Datasets

Training datasets. For training we used three datasets, namely ADE20K [38], Cityscapes [11], COCO-Stuff [8]. ADE20K [38] offers a diverse array of scenes, Cityscapes [11] focuses on urban environments, and COCO-Stuff [8] provides a wide range of objects and complex scenes, together forming a robust foundation for training our models.

Evaluation datasets. For evaluation we use 5 out-of-distribution datasets containing unseen classes that do not appear during training: Dental-teeth segmentation [17], Dental-jaw segmentation [1], Flood dataset [26], LIVECell dataset [12] containing cell segmentation, SHREC dataset [30] containing potholes and road cracks segmentation.

Additionally, in order to assess performance on classes seen during training, experiments were conducted on the PASCAL-5i dataset [29] which is derived from the PascalVOC dataset [14]. It contains 20 classes, evenly divided into four folds (F1-4). It is crucial to clarify that while our final model was not directly trained on either the PascalVOC or PASCAL-5i datasets, classes similar to those contained in these datasets are present in the training data. Following previous methods, we report mean intersection-over-union (mIoU) metric for all experiments.

4.2 Ablation studies

In this section, our objective is to address two important questions. Firstly, we investigate how the selection of the support set influences performance. To provide a comprehensive understanding, we present results for both In-distribution and Out-of-distribution datasets, offering a broad perspective on our method's effectiveness across different scenarios. Secondly, we explore the impact of finetuning XMem on image datasets. This is particularly of interest since as previously noted, XMem was originally trained on video frames, where changes in scenery are typically more gradual and less pronounced compared to still image datasets. Understanding how this transition to image datasets influences performance is therefore essential.

For both abaltion studies, the experimental results are categorized into two groups: In-distribution datasets and Out-of-distribution datasets. In-distribution refers to the evaluation splits of Cityscapes and COCO-Stuff datasets, while Out-of-distribution refers to datasets that were unseen during training, specifically Dental-teeth, Dental-jaw, Flood, LIVECell and SHREC. Mean results over the datasets are presented for both categories, with explicit results per each dataset available in the Appendix.

4.2.1 Support set selection

We conduct support set selection experiments, presenting results for seven methods in Table 1. The tested methods are Universeg [7], PerSAM [37], STCN [10], SegGPT [33], SegGPT with logits averaging, XMem and XMem+T. Each method improves with support set selection, some by over 14%. CLIP [25] features were used for all support set experiments.

4.2.2 Training

We explore the impact of training XMem on a large dataset of images, as detailed in Section 3.2. The results are presented in Table 2 on both In-distribution and Out-of-distribution datasets, while detailed results per dataset are presented in the Appendix. As it can be seen, training on a large

Table 1: **Impact of support set selection (s^3) on In-distribution and Out-of-distribution datasets.** We report mIoU for different ICL methods with and without support set selection (s^3) using different support set sizes. + T performs image training. † our proposed solution of averaging over logits.

		In-distribution datasets		Out-of-distribution datasets	
		Support set size		Support set size	
Method	s^3	1	10	1	10
Universeg [7]	-	5.75	11.89	22.06	44.84
Universeg [7]	✓	7.65 (↑ 1.9)	15.51 (↑ 3.62)	26.62 (↑ 4.56)	51.22 (↑ 6.38)
PerSAM [37]	-	17.65	-	24.7	-
PerSAM [37]	✓	21.79 (↑ 4.14)	-	31.2 (↑ 6.5)	-
STCN [10]	-	11.34	13.28	52.39	52.92
STCN [10]	✓	18.74 (↑ 7.4)	18.62 (↑ 5.34)	66.11 (↑ 13.71)	56.40 (↑ 3.48)
SegGPT [33]	-	31.31	36.56	49.77	52.38
SegGPT [33]	✓	36.18 (↑ 4.97)	38.0 (↑ 1.44)	63.09 (↑ 13.32)	63.17 (↑ 10.79)
SegGPT logit†	-	31.31	40.52	49.77	58.84
SegGPT logit†	✓	36.18 (↑ 4.87)	43.86 (↑ 3.34)	63.09 (↑ 13.32)	65.12 (↑ 6.28)
XMem [9]	-	12.84	17.83	54.56	65.89
XMem [9]	✓	20.07 (↑ 7.23)	25.36 (↑ 7.63)	68.97 (↑ 14.41)	73.34 (↑ 7.45)
XMem + T	-	17.81	26.34	54.52	65.6
XMem + T	✓	25.04 (↑ 7.23)	32.74 (↑ 6.4)	67.32 (↑ 12.8)	71.83 (↑ 6.23)

Table 2: **Impact of finetuning on XMem performance**. We report mIoU on In-distribution and Out-of-distribution datasets and compare XMem with and without training with image datasets (+ T), for different support set sizes. The results are presented with support set selection.

	In-distribution datasets		Out-of-distribution datasets	
	Support set size		Support set size	
Method	1	10	1	10
XMem [9]	20.07	25.36	68.97	73.34
XMem + T	25.04 (↑ 4.96)	32.74 (↑ 7.38)	67.32 (↓ 1.65)	71.83 (↓ 1.51)

dataset of images enhances performance on In-distribution datasets while slightly decreasing it on Out-of-distribution datasets. However, the trained model (XMem + T) represents a favorable compromise, offering improved performance on a diverse range of classes. It excels particularly on classes it has seen, making it a viable choice for those seeking a balance in performance across various datasets.

Table 3: **Results on PASCAL-5i**. We report results on PASCAL-5i on all folds (F0, F1, F2, F3) and compare Xmem with support set selection (s^3) to other ICL methods. SegGPT [33] (in grey) is not comparable because it was trained on PascalVOC. † our proposed solution of averaging over logits.

	Support set size 1					Support set size 10				
Method	F0	F1	F2	F3	Mean	F0	F1	F2	F3	Mean
Visual prompting [5]	27.83	30.44	26.15	24.25	27.16	–	–	–	–	–
SegGPT [33]	66.92	73.55	76.90	72.44	72.45	68.89	76.70	81.47	76.50	75.89
Universeg [7]	12.66	13.53	12.43	12.60	12.80	21.52	22.34	22.14	22.31	22.08
PerSAM [37]	48.13	43.61	**52.87**	42.72	46.83	-	-	-	-	-
STCN [10]	27.14	28.96	27.20	25.40	27.17	31.10	25.79	28.46	20.44	26.45
SegGPT logits-avg†+s^3	70.09	76.91	78.70	74.55	75.06	69.73	78.85	81.14	76.56	76.57
XMem + s^3	47.41	45.49	43.23	38.19	43.58	53.82	52.77	51.60	41.68	49.97
XMem + T + s^3	**49.56**	**50.78**	49.28	**46.26**	**48.97**	**58.65**	**60.16**	**60.25**	**53.65**	**58.18**

Table 4: **Generalisation results on Out-of-distribution datasets**. We report mIoU on 5 out-of-distribution datasets and compare XMem and XMem finetuned with support set selection (s^3) to other ICL methods using support set sizes 1 and 10. † our proposed solution of averaging over logits.

	Dental-teeth		Dental-jaw		Flood		LIVECell		SHREC	
Method	1	10	1	10	1	10	1	10	1	10
SegGPT [33]	72.76	76.26	74.04	**83.70**	38.70	39.02	40.58	41.30	22.78	21.63
Universeg [7]	40.67	67.23	42.20	73.46	8.77	21.45	15.53	41.97	3.11	20.07
PerSAM [37]	21.37	-	47.73	-	20.18	-	24.06	-	10.14	-
STCN [10]	77.35	67.72	73.10	67.96	34.30	41.96	54.39	57.74	22.82	29.21
SegGPT logits-avg† + s^3	74.53	77.51	79.54	69.15	66.05	64.81	59.94	70.45	35.41	43.67
XMem + s^3	**79.93**	**82.88**	**80.75**	83.58	**67.54**	**72.82**	**78.68**	**80.57**	37.93	**46.84**
XMem + T + s^3	76.87	80.97	77.93	83.13	67.24	72.50	76.11	77.90	**38.45**	44.67

4.3 Comparison with other ICL methods

We evaluated our final method (XMem + T + support set selection) against several methods, namely: Visual prompting, Universeg [7], PerSAM [37], STCN [10], and SegGPT [33]. It is important to highlight a significant observation made during our experiments: SegGPT method typically averages features when the support set contains more than one example, while our experiments revealed a superior approach. Specifically, we found that processing one example at a time and averaging logits across the entire support set yields improved results — sometimes enhancing performance by over 10% — especially for Out-of-distribution datasets (as shown in Table 4). Further, results are presented using both the original SegGPT and our modified version that implements logit averaging. We start by showing results on Pascal-5i in Table 3.

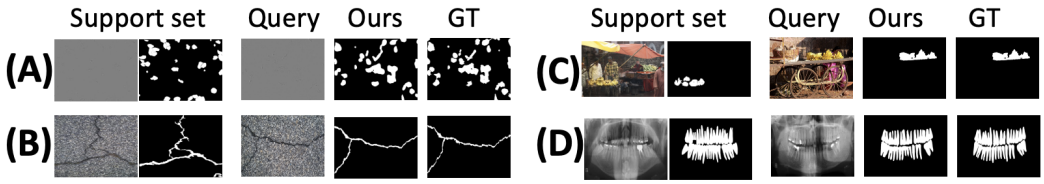

Figure 3: Qualitative results on (A) LIVECell [12], (B) SHREC [30], (C) COCO-Stuff [8], (D) Dental-teeth [17]. Ours represents XMem with support set selection. The same model weights are used to segment a variety of classes, such as biological cells, bananas, dental radiography and road surface cracks.

Generalisation Results. We present results on five out-of-distribution datasets in Table 4. XMem with support set selection shows superior generalisation capabilities, notably improving performance on the Flood and LIVECell datasets.

Qualitative results In Figure 3 we present qualitative results on four datasets. We refer to XMem with support set selection as ours. We observe that by using the same model, specifically XMem + s^3, we are able to segment various classes.

5 Conclusion

Our study targets visual in-context learning (ICL) specifically for binary semantic segmentation tasks. We are the first to explore the application of Video Object Segmentation (VOS) methodologies to in-context learning. Through extensive analysis, we demonstrate that our adapted VOS methodology offers superior generalization to unseen classes, providing a more robust and versatile solution than existing ICL methods.

In addition, we also explore the significance of support set selection in enhancing the performance of visual in-context learning (ICL). We found that a strategic approach to choosing support sets can lead to considerable improvements in performance. This becomes particularly notable as we observed that such selection of support sets yielded enhanced results across all seven ICL methods we tested.

References

[1] Amir Hossein Abdi, Shohreh Kasaei, and Mojdeh Mehdizadeh. Automatic segmentation of mandible in panoramic x-ray. *Journal of Medical Imaging*, 2(4):044003, 2015.

[2] Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. In-context examples selection for machine translation. *arXiv preprint arXiv:2212.02437*, 2022.

[3] Mostafa M Amin, Erik Cambria, and Björn W Schuller. Will affective computing emerge from foundation models and general artificial intelligence? a first evaluation of chatgpt. *IEEE Intelligent Systems*, 38(2):15–23, 2023.

[4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[5] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022.

[6] 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.

[7] Victor Ion Butoi, Jose Javier Gonzalez Ortiz, Tianyu Ma, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Universeg: Universal medical image segmentation. *arXiv preprint arXiv:2304.06131*, 2023.

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[9] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022.

[10] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[12] Christoffer Edlund, Timothy R Jackson, Nabeel Khalid, Nicola Bevan, Timothy Dale, Andreas Dengel, Sheraz Ahmed, Johan Trygg, and Rickard Sjögren. Livecell—a large-scale dataset for label-free live cell segmentation. *Nature methods*, 18(9):1038–1045, 2021.

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[14] 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.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Serdar HELLİ and Andaç HAMAMCI. Tooth instance segmentation on panoramic dental radiographs using u-nets and morphological processing. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 10(1):39–50.

[18] Zixian Huang, Jiaying Zhou, Gengyang Xiao, and Gong Cheng. Enhancing in-context learning with answer feedback for multi-span question answering. *arXiv preprint arXiv:2306.04508*, 2023.

[19] 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. *arXiv preprint arXiv:2304.02643*, 2023.

[20] 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.

[21] Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. Few-shot in-context learning for knowledge base question answering. *arXiv preprint arXiv:2305.01750*, 2023.

[22] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. *CVPR*, 2020.

[23] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.

[24] Zhihe Lu, Sen He, Xiatian Zhu, Li Zhang, Yi-Zhe Song, and Tao Xiang. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8741–8750, 2021.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[26] Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021.

[27] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

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

[30] Elia Moscoso Thompson, Andrea Ranieri, Silvia Biasotti, Miguel Chicchon, Ivan Sipiran, Minh-Khoi Pham, Thang-Long Nguyen-Ho, Hai-Dang Nguyen, and Minh-Triet Tran. Shrec 2022: Pothole and crack detection in the road pavement using images and rgb-d data. *Computers & Graphics*, 107:161–171, 2022.

[31] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *European Conference on Computer Vision*, 2020.

[32] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023.

[33] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*, 2023.

[34] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[35] Yong Yang, Qiong Chen, Yuan Feng, and Tianlin Huang. Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7131–7140, 2023.

[36] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems*, 34:2491–2502, 2021.

[37] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. *arXiv preprint arXiv:2305.03048*, 2023.

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

A Datasets

In Table 5 we show how many images are present in the training and evaluation for each of the datasets used throughout the paper. As specified in the main paper for training we used: ADE20K [38], Cityscapes [11], COCO-Stuff [8]. We also conducted assessments on the evaluation set of those datasets, with results presented in Table 7. Additional evaluations were performed on Dental-teeth segmentation [17], Dental-jaw segmentation [1], Flood dataset [26], LIVECell dataset [12] containing cell segmentation, SHREC dataset [30] containing potholes and road cracks segmentation and on PASCAL-5i dataset.

Table 5: **Datasets used for training and evaluation.** * PASCAL-5i dataset has 20 classes split equally in 4 folds.

	Training		Evaluation	
Dataset name	#classes	#pairs	#classes	#pairs
ADE20K	130	165120	0	0
COCO-Stuff	171	965042	171	42095
Cityscapes	19	34723	19	6006
PASCAL-5i	0	0	20*	1964
Dental-teeth	0	0	1	47
Dental-jaw	0	0	2	47
Flood	0	0	9	1740
LIVECell	0	0	1	1512
SHREC	0	0	3	290

Dental-teeth segmentation [17], Dental-jaw segmentation [1], Flood dataset [26], LIVECell dataset [12] containing cell segmentation, SHREC dataset [30] containing potholes and road cracks segmentation.

B Additional Results

Below we present the detailed results of all the methods and on all datasets. In Tab. 6 we present the results of all tested methods on PASCAL-5i. In Table 7 we present results of all methods on the evaluation split of all training datasets. Further, we present additional results in Table 8 for Out-of-distribution datasets.

Table 6: **Results on PASCAL-5i**. We report results on PASCAL-5i on all folds (F0, F1, F2, F3) and compare Xmem to other ICL methods. We extend Table 3 and present here all variants for support set selection (s^3). SegGPT [33] (in grey) is not comparable because it was trained on PascalVOC. † our proposed solution of averaging over logits.

		Support set size 1					Support set size 10				
Method	s^3	F0	F1	F2	F3	Mean	F0	F1	F2	F3	Mean
Visual prompting [5]	-	27.83	30.44	26.15	24.25	27.16	–	–	–	–	–
SegGPT [33]	-	66.92	73.55	76.90	72.44	72.45	68.89	76.70	81.47	76.50	75.89
SegGPT [33]	✓	70.09	76.91	78.70	74.55	75.06	71.22	79.26	81.88	77.02	77.34
Universeg [7]	-	12.66	13.53	12.43	12.60	12.80	21.52	22.34	22.14	22.31	22.08
Universeg [7]	✓	17.54	16.97	16.50	16.97	16.99	32.15	32.97	30.65	29.06	31.21
PerSAM [37]	-	48.13	43.61	**52.87**	42.72	46.83	-	-	-	-	-
PerSAM [37]	✓	59.61	56.24	55.41	49.44	55.17	-	-	-	-	-
STCN [10]	-	27.14	28.96	27.20	25.40	27.17	31.10	25.79	28.46	20.44	26.45
STCN [10]	✓	44.64	43.32	42.40	37.13	41.87	40.25	32.54	35.38	22.42	32.65
SegGPT logits-avg	-	66.92	73.44	76.24	73.30	72.65	68.43	78.42	80.65	77.30	76.46
SegGPT logits-avg†	✓	70.09	76.91	78.70	74.55	75.06	69.73	78.85	81.14	76.56	76.57
XMem	-	34.10	32.34	29.05	28.52	31.00	42.57	41.71	38.59	33.55	39.10
XMem	✓	47.41	45.49	43.23	38.19	43.58	53.82	52.77	51.60	41.68	49.97
XMem + FT	-	34.20	38.04	34.34	34.28	35.22	47.23	50.82	49.87	46.30	48.56
XMem + FT	✓	**49.56**	**50.78**	49.28	**46.26**	**48.97**	**58.65**	**60.16**	**60.25**	**53.65**	**58.18**

Table 7: **Results on Cityscapes, and COCO-Stuff.** We report mIoU for different ICL methods on the evaluation splits of the training datasets, with and without support set selection (s^3), and using different support set sizes. + T performs training on image datasets. † averaging over logits and support set selection.

		Cityscapes		COCO-Stuff	
		Support set size		Support set size	
Method	s^3	1	10	1	10
SegGPT [33]	-	35.43	37.27	27.19	35.84
SegGPT [33]	✓	36.74	37.79	35.62	38.21
Universeg [7]	-	6.45	13.83	5.05	9.94
Universeg [7]	✓	7.50	15.64	7.80	15.38
PerSAM [37]	-	16.04	-	19.25	-
PerSAM [37]	✓	16.64	-	26.93	-
STCN [10]	-	13.03	14.53	9.65	12.03
STCN [10]	✓	18.17	21.40	19.30	15.84
segGPT logit-avg†	-	35.43	41.88	27.19	39.16
SegGPT logit-avg†	✓	36.74	43.31	35.62	44.41
XMem	-	15.80	21.94	9.87	13.71
XMem	✓	20.31	26.37	19.83	24.55
XMem + FT	-	21.51	30.81	14.10	21.86
XMem + FT	✓	25.00	33.59	25.08	31.88

Table 8: **Generalisation results on Out-of-distribution datasets**. We report mIoU on 5 out-of-distribution datasets and compare XMem and XMem image training (+ T) to other ICL methods using different support set sizes. We extend Table 4 and present here all variants for support set selection (s^3). † our proposed solution of averaging over logits.

		Dental-teeth		Dental-jaw		Flood		LIVECell		SHREC	
Method	s^3	1	10	1	10	1	10	1	10	1	10
SegGPT [33]	-	72.76	76.26	74.04	83.70	38.70	39.02	40.58	41.30	22.78	21.63
SegGPT [33]	✓	74.53	76.02	79.54	83.53	66.05	63.84	59.94	55.57	35.41	36.88
Universeg [7]	-	40.67	67.23	42.20	73.46	8.77	21.45	15.53	41.97	3.11	20.07
Universeg [7]	✓	40.17	69.44	44.70	75.42	20.26	37.33	22.95	51.80	5.03	22.12
PerSAM [37]	-	21.37	-	47.73	-	20.18	-	24.06	-	10.14	-
PerSAM [37]	✓	22.19	-	46.81	-	40.78	-	25.85	-	18.22	-
STCN [10]	-	77.35	67.72	73.10	67.96	34.30	41.96	54.39	57.74	22.82	29.21
STCN [10]	✓	78.67	66.64	79.44	67.45	67.37	58.16	72.55	55.59	32.50	34.16
SegGPT logits-avg†	-	72.76	77.33	74.04	66.28	38.70	56.28	40.58	54.87	22.78	39.46
SegGPT logits-avg†	✓	74.53	77.51	79.54	69.15	66.05	64.81	59.94	70.45	35.41	43.67
XMem	-	78.51	82.88	73.93	**85.19**	32.50	48.30	65.14	75.79	22.72	37.27
XMem	✓	**79.93**	**82.88**	**80.75**	83.58	**67.54**	**72.82**	**78.68**	**80.57**	37.93	**46.84**
XMem + FT	-	75.94	80.65	70.72	84.11	36.83	52.08	60.39	71.85	28.71	39.30
XMem + FT	✓	76.87	80.97	77.93	83.13	67.24	72.50	76.11	77.90	**38.45**	44.67