A Partially Supervised Reinforcement Learning Framework for Visual Active Search: Supplementary Material

A Policy Network Architecture and Hyperparameter Details

Recall that the policy network π is composed of two parts : (1) a task specific prediction module, and (2) a task-agnostic search module. The task specific prediction module consists of an encoder $e(x;\eta)$ that maps the aerial image x to a low-dimensional latent feature representation z, and a grid prediction network $p(z,o;\kappa)$ that predicts the probabilities of grids containing a target by leveraging the latent semantic feature z and the outcomes of previous search queries o. Note that the task specific prediction module is represented as $f(x,o,\theta) = p(z = e(x;\eta), o;\kappa)$, where $\theta = (\eta,\kappa)$. Following [6], we use frozen ResNet-34, pre-trained on ImageNet, followed by a learnable 1×1 convolution layer with a ReLU activation as a feature extraction component of the task specific prediction module that we refer as encoder $e(.)$. We then combine the latent semantic feature z with the previous query information o. We apply the tiling operation in order to convert o into a representation with the same dimensions as the extracted features z, enabling us to effectively apply channel-wise concatenation of latent image feature and auxiliary state feature while preserving the grid specific spatial and query related information. This combined representation is then fed to a grid prediction network comprises of a 1×1 convolution layer, flattening, and a MLP block consists of 2 fully connected layer with ReLU activations. Note that the output of grid prediction network is of dimension N. We finally apply sigmoid activation to each output neuron to convert them into a probability value representing the probability of the grids containing target. The proposed policy architecture is depicted in figure 2 of the main paper.

We re-shape the output of task specific prediction module by converting it back from 1D to 2D of shape $(m \times n) = N$ before feeding it to the task agnostic search module $g(.)$ that takes the following three inputs: (1) the reshaped 2D output of the task specific prediction module, which is the probabilities of grids containing target; (2) the remaining search budget B, which is a scalar but we apply tiling to the scalar budget B to transform it to match the size of the reshaped 2D output of the task specific prediction module; (3) we also apply the tiling operation to o in a way that allows us to concatenate the features (z, o, B) along the channels dimension to finally obtain the combined representation that serves as a input to task agnostic search module. The task agnostic search module is composed of a flattening, a MLP block consists of 2 fully connected layer with ReLU activations, and a final softmax layer to convert the output to a probability distribution that guides us in selecting the grid to query next.

In Table 5, we detail the architecture of task specific prediction module (f) of PSVAS policy network. In Table 6, we detail the architecture of task agnostic search module (g) of PSVAS policy network. Note that, the task specific prediction module and task agnostic search module remains unchanged in MPS-VAS framework.

Table 5: Task Specific Prediction Module Architecture with number of grid cell $N = (m \times n)$

Layers	**Configuration**	**o/p Feature Map size**
Input	RGB Image	3 × 3500 × 3500
Encoder	ResNet-34	512 × 14 × 14
Conv1	Channel:N; kernel size:1 × 1	N × 14 × 14
2D MaxPool	Pooling size:2 × 2	N × 7 × 7
Tile1	Grid State (o)	$N \times 7 \times 7$
Channelwise Concat	Conv1,Tile1	$(2N) \times 7 \times 7$
Conv2	Channel:3; kernel size: 1×1	3 × 7 × 7
Flattened	Conv2	147
FC1+ReLU	$(147- > 2N)$	2N
FC2+Sigmoid	$(2N- > N)$	N

Table 6: Task Agnostic Search Module Architecture with number of grid cell $N = (m \times n)$

Layers	Configuration	o/p Feature Map size
Input 1	2D Reshape of Task Specific Prediction Module Output	$1 \times m \times n$
Input 2: Tile2	Grid State (o)	$1 \times m \times n$
Input 3: Tile3	Query Budget Left (B)	$1 \times m \times n$
Input: Channelwise Concat	Input 1, Input 2, Input 3	$(3) \times m \times n$
Flattened	Input: Channelwise Concat	$K = (3) \times m \times n$
FC1+ReLU	$(K- > 2N)$	2N
FC2+Softmax	$(2N- > N)$	N

In MPS-VAS-MQ framework, the network architecture of task specific prediction module remains unaltered, but the additional dependence of task agnostic search module (g) on ψ enforce a slight modification of its architecture as detailed in Table 7.

Table 7: Task Agnostic Search Module Architecture in multi query setting with number of grid cell $N = (m \times n)$

Layers	Configuration	o/p Feature Map size
Input 1	2D Reshape of Task Specific Prediction Module Output	$1 \times m \times n$
Input 2: Tile2	Grid State (o)	$1 \times m \times n$
Input 3: Tile3	Query Budget Left (B)	$1 \times m \times n$
Input 4: Tile4	Encoded Locations of the queried Grid cells (ψ)	$1 \times m \times n$
Input: Channelwise Concat	Input 1, Input 2, Input 3, Input 4	$(4) \times m \times n$
Flattened	Input: Channelwise Concat	$D = (4) \times m \times n$
FC1+ReLU	$(D- > 2N)$	2N
FC2+Softmax	$(2N- > N)$	N

We use a learning rate of 10^{-4}, batch size of 16, number of training epochs 200, and the Adam optimizer to train the policy network in all experimental settings. During Inference, in all experimental settings, we update the parameters of task specific prediction module f after each query step using a learning rate of 10^{-4} and the Adam optimizer. We use 1 NVidia A100 and 3 GeForce GTX 1080Ti GPU servers for all our experiments.

B Results with Uniform Query Cost

B.1 Single Query Setting

Here we present the results by considering a setting with a single query resource and query costs $c(i, j) = 1$ for all i, j, where $\mathcal{C}$ is the number of queries. We evaluate PSVAS and MPS-VAS on the xView dataset with varying search budget $\mathcal{C} \in \{12, 15, 18\}$ and the number of grid cells $N = 49$. We train the policy with *small car* as the target and test the performance of the policy with the following target classes : *Small Car (SC)*, *Helicopter*, *Sail Boat (SB)*, *Construction Cite (CC)*, *Building*, and *Helipad*. The results are presented in Table 8. We observe noticeable improvement in performance of the proposed PSVAS approach compared to all baselines in each different target setting, ranging from approximately 0.50 to 52.0% relative to the most competitive E2EVAS baseline. In Table 9, we report the results on DOTA dataset with $N = 64$. In this setting, we train the policy with *large vehicle* as the target and evaluate the performance with the following target classes : *Ship*, *large vehicle* (LV), *Harbor*, *Helicopter*, *Plane*, and *Roundabout*. Here, we notice significant improvement in performance of PSVAS compared to all the baselines including E2EVAS, ranging from approximately 3.5 to 25.0%. The effectiveness of the PSVAS framework becomes evident as it allows us to efficiently update the task-specific prediction module f by leveraging the crucial supervised information. We also observe a consistent trend, i.e., the performance of MPS-VAS is significantly better than PSVAS across different target settings, ranging from approximately 0.6 to 60.0%. The significance of the MPS-VAS framework becomes apparent when deploying visual active search in scenarios where the search tasks differ substantially from those encountered during training.

Table 8: **ANT** comparisons when trained with *small car* as target on xView in single-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
RS	0.41	0.52	0.65	0.62	0.83	0.93	4.74	6.05	7.11
GC	0.44	0.59	0.78	0.73	0.92	0.99	5.45	6.53	7.65
GS [15]	0.47	0.61	0.84	0.78	0.96	1.03	5.68	6.87	8.01
AL [13]	0.43	0.59	0.77	0.72	0.90	0.97	5.44	6.53	7.63
AS [9]	0.44	0.57	0.75	0.70	0.89	0.96	5.32	6.38	7.44
E2EVAS [6]	0.50	0.63	0.92	0.83	1.06	1.10	7.29	8.78	10.14
OnlineTTA[6]	0.50	0.64	0.93	0.84	1.06	1.11	7.29	8.79	10.15
PSVAS	**0.91**	**0.95**	**1.08**	**0.97**	**1.13**	**1.37**	**7.30**	**8.81**	**10.28**
MPS-VAS	**1.04**	**1.13**	**1.21**	**1.23**	**1.50**	**1.74**	**7.32**	**8.83**	**10.33**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
RS	1.19	1.54	1.81	3.62	4.57	5.51	0.38	0.47	0.61
GC	1.42	1.86	2.19	4.06	4.98	6.03	0.51	0.65	0.83
GS [15]	1.61	2.01	2.33	4.59	5.54	6.71	0.56	0.74	0.96
AL [13]	1.41	1.85	2.17	4.03	4.96	6.02	0.51	0.63	0.82
AS [9]	1.40	1.74	2.09	3.96	4.92	5.97	0.47	0.59	0.77
E2EVAS [6]	1.74	2.10	2.46	5.80	7.02	8.15	0.90	1.06	1.23
OnlineTTA[6]	1.75	2.12	2.46	5.81	7.03	8.15	0.91	1.06	1.23
PSVAS	**1.86**	**2.25**	**2.61**	**5.94**	**7.10**	**8.19**	**1.02**	**1.09**	**1.26**
MPS-VAS	**1.97**	**2.35**	**2.76**	**5.99**	**7.16**	**8.24**	**1.07**	**1.16**	**1.37**

Table 9: **ANT** comparisons when trained with *large vehicle* as target on DOTA in single-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
Random	2.41	3.02	3.95	3.40	4.03	5.14	3.17	3.93	4.78
GC	2.82	3.44	4.27	3.87	4.59	5.55	3.48	4.25	4.98
GS[15]	2.96	3.59	4.48	3.99	4.77	5.67	3.62	4.40	5.07
AL[13]	2.81	3.42	4.26	3.85	4.54	5.51	3.47	4.25	4.97
AS[9]	2.57	3.27	4.03	3.61	4.12	5.26	3.35	4.16	4.92
E2EVAS[6]	3.57	4.42	5.15	6.30	7.65	8.90	4.28	5.21	6.09
OnlineTTA[6]	3.57	4.43	5.15	6.31	7.67	8.90	4.30	5.22	6.10
PSVAS	**3.60**	**4.51**	**5.23**	**6.50**	**7.86**	**9.22**	**4.61**	**5.72**	**6.87**
MPS-VAS	**3.79**	**4.75**	**5.58**	**6.51**	**7.88**	**9.24**	**4.90**	**6.23**	**7.38**
	Test with Helicopter as Target			Test with Plane as Target			Test with Roundabout as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
Random	0.66	0.73	0.82	2.91	3.94	4.74	2.66	3.59	4.37
GC	0.71	0.82	0.89	3.22	4.35	5.07	2.93	3.81	4.59
GS[15]	0.75	0.87	0.97	3.47	4.56	5.25	2.99	3.96	4.73
AL[13]	0.70	0.81	0.88	3.22	4.34	5.07	2.93	3.79	4.59
AS[9]	0.68	0.78	0.86	3.16	4.21	4.97	2.82	3.74	4.51
E2EVAS[6]	0.78	0.96	1.18	4.02	5.07	5.90	4.00	5.05	5.88
OnlineTTA[6]	0.78	0.97	1.19	4.02	5.07	5.91	4.01	5.06	5.88
PSVAS	**0.95**	**1.21**	**1.49**	**4.33**	**5.32**	**6.44**	**4.33**	**5.36**	**6.41**
MPS-VAS	**1.10**	**1.37**	**1.67**	**4.52**	**5.58**	**6.75**	**4.51**	**5.56**	**6.73**

B.2 Multi Query Setting

In Table 10, we present the results of MPS-VAS-MQ and compare its performance with MPS-VAS-TOPK with varying search budget $\mathcal{C} \in \{12, 15, 18\}$ and the number of grid cell N=49. Here, we train the policy with *small car* as the target and evaluate the performance of the policy with the following target classes : *Small Car* (SC), *Helicopter*, *Sail Boat* (SB), *Construction Cite* (CC), *Building*, and *Helipad*. In table 11, we present similar results with the number of grid cell $N = 64$. In this setting, we train the policy with *Large Vehicle* as the target and evaluate the policy with the following target classes: *Ship*, *Large Vehicle* (LV), *Harbor*, *Helicopter*, *Plane*, and *Roundabout* (RB).

We consider $K = 3$ in all these experiments. We observe a consistent improvement in performance of MPS-VAS-MQ over MPS-VAS-TOPK across different target setting, ranging from approximately 0.1 to 15%. The experimental results indicate that there are additional benefits in learning to capture the interdependence in greedy search decisions.

Table 10: **ANT** comparisons when trained with *small car* as target on xView in multi-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C} = 12$	$\mathcal{C} = 15$	$\mathcal{C} = 18$	$\mathcal{C} = 12$	$\mathcal{C} = 15$	$\mathcal{C} = 18$	$\mathcal{C} = 12$	$\mathcal{C} = 15$	$\mathcal{C} = 18$
MPS-VAS-TOPK	0.71	0.85	1.04	1.10	1.23	1.47	7.07	8.60	9.98
MPS-VAS-MQ	**0.75**	**0.88**	**1.08**	**1.14**	**1.41**	**1.53**	**7.31**	**8.81**	**10.21**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C} = 12$	$\mathcal{C} = 15$	$\mathcal{C} = 18$	$\mathcal{C} = 12$	$\mathcal{C} = 15$	$\mathcal{C} = 18$	$\mathcal{C} = 12$	$\mathcal{C} = 15$	$\mathcal{C} = 18$
MPS-VAS-TOPK	1.89	2.09	2.50	5.78	6.92	7.98	0.82	0.93	1.10
MPS-VAS-MQ	**1.95**	**2.27**	**2.68**	**5.97**	**7.09**	**8.16**	**1.03**	**1.09**	**1.23**

Table 11: **ANT** comparisons when trained with *large vehicle* as target on DOTA in multi-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$
MPS-VAS-TOPK	3.72	4.66	5.49	6.09	7.29	8.54	4.76	6.14	7.31
MPS-VAS-MQ	**3.74**	**4.69**	**5.54**	**6.36**	**7.64**	**8.79**	**4.78**	**6.20**	**7.32**
	Test with Helicopter as Target			Test with Plane as Target			Test with RB as Target		
Method	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$
MPS-VAS-TOPK	0.88	1.05	1.24	3.95	5.48	6.69	4.32	5.45	6.45
MPS-VAS-MQ	**0.90**	**1.06**	**1.30**	**4.02**	**5.49**	**6.73**	**4.39**	**5.47**	**6.49**

C Results with Different Number of grid cells

Here, we present the results of PSVAS and MPS-VAS and compare the performance with the most competitive E2EVAS approach for different choices of N.

C.1 Results with Number of Grid cell $N = 99$

In this setting, we train the policy with *small car* as the target and evaluate the performance of the policy with the following target classes : *Small Car* (SC), *Helicopter*, *Sail Boat* (SB), *Construction Cite* (CC), *Building*, and *Helipad*. In Table 12, we present the results with *Manhattan distance based query cost* in single query setting. The similar results with multi query setting are presented in Table 13. In Table 14 and 15, we present the results with *uniform query cost* in single and multi query setting respectively. We notice a very similar trend in performance as observed in the settings with other choices of N. Specifically, We observe PSVAS significantly outperforms E2EVAS across different target settings, and MPS-VAS further improves the search performance universally. These results highlights the effectiveness of our proposed PSVAS and MPS-VAS framework for visual active search in practical scenarios when search tasks differ from those that are used for policy training.

C.2 Results with Number of Grid cell $N = 36$

In this setting, we train the policy with *large vehicle* as the target and evaluate the performance with the following target classes : *Ship*, *large vehicle* (LV), *Harbor*, *Helicopter*, *Plane*, and *Roundabout*. In Table 16, we present the results with Manhattan distance based query cost in single query setting. The results with multi query setting are presented in Table 17. In Table 18 and 19, we present the the results with uniform query cost in single and multi query setting respectively. We observe a consistent performance trend across various target settings. Specifically, PSVAS consistently outperforms E2EVAS in different target settings, and the introduction of MPS-VAS further enhances the search performance across the board. These results emphasize the effectiveness of our proposed PSVAS and

Table 12: **ANT** comparisons when trained with *small car* as target on xView in single-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
RS	0.01	0.09	0.14	0.23	0.34	0.61	1.41	2.51	3.84
E2EVAS [6]	0.17	0.30	0.39	0.65	1.03	1.34	3.32	5.37	7.05
OnlineTTA[6]	0.17	0.31	0.40	0.66	1.03	1.34	3.32	5.39	7.07
PSVAS	**0.39**	**0.48**	**0.65**	**0.71**	**1.07**	**1.35**	**4.31**	**6.97**	**9.12**
MPS-VAS	**0.45**	**0.55**	**0.69**	**0.75**	**1.08**	**1.37**	**4.42**	**7.18**	**9.35**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
RS	0.32	0.56	0.87	1.10	2.15	2.96	0.12	0.19	0.29
E2EVAS [6]	0.61	1.03	1.41	2.72	4.42	5.78	0.39	0.44	0.56
OnlineTTA[6]	0.63	1.04	1.41	2.72	4.43	5.79	0.39	0.45	0.56
PSVAS	**0.98**	**1.72**	**2.19**	**3.12**	**5.01**	**6.40**	**0.46**	**0.59**	**0.74**
MPS-VAS	**1.01**	**1.77**	**2.28**	**3.34**	**5.31**	**6.74**	**0.51**	**0.66**	**0.86**

Table 13: **ANT** comparisons when trained with *small car* as target on xView in multi-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
MPS-VAS-TOPK	0.40	0.51	0.62	0.69	0.98	1.30	4.29	6.84	8.66
MPS-VAS-MQ	**0.42**	**0.53**	**0.66**	**0.71**	**1.03**	**1.32**	**4.33**	**6.95**	**8.78**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
MPS-VAS-TOPK	0.96	1.53	2.12	3.19	5.09	6.47	0.45	0.59	0.77
MPS-VAS-MQ	**0.98**	**1.65**	**2.17**	**3.25**	**5.12**	**6.55**	**0.47**	**0.61**	**0.82**

Table 14: **ANT** comparisons when trained with *small car* as target on xView in single-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
RS	0.22	0.31	0.38	0.48	0.55	0.63	3.43	4.25	4.97
E2EVAS [6]	0.31	0.39	0.43	0.80	1.05	1.30	5.23	6.37	7.41
OnlineTTA[6]	0.31	0.40	0.44	0.80	1.06	1.31	5.24	6.38	7.43
PSVAS	**0.43**	**0.48**	**0.51**	**0.83**	**1.09**	**1.33**	**5.34**	**6.41**	**7.52**
MPS-VAS	**0.47**	**0.50**	**0.54**	**0.84**	**1.11**	**1.39**	**5.44**	**6.69**	**7.75**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
RS	0.78	1.02	1.17	3.12	3.61	4.45	0.25	0.33	0.41
E2EVAS [6]	0.98	1.29	1.47	4.61	5.64	6.55	0.44	0.46	0.56
OnlineTTA[6]	0.99	1.32	1.50	4.62	5.64	6.56	0.45	0.47	0.56
PSVAS	**1.28**	**1.64**	**1.86**	**4.74**	**5.72**	**6.75**	**0.53**	**0.59**	**0.78**
MPS-VAS	**1.39**	**1.69**	**2.05**	**4.81**	**5.93**	**6.96**	**0.61**	**0.66**	**0.83**

Table 15: **ANT** comparisons when trained with *small car* as target on xView in multi-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
MPS-VAS-TOPK	0.41	0.43	0.48	0.78	1.01	1.26	4.91	6.07	7.02
MPS-VAS-MQ	**0.42**	**0.46**	**0.51**	**0.81**	**1.05**	**1.32**	**5.02**	**6.21**	**7.18**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
MPS-VAS-TOPK	1.22	1.41	1.82	4.29	5.63	6.59	0.54	0.59	0.78
MPS-VAS-MQ	**1.26**	**1.53**	**1.98**	**4.38**	**5.74**	**6.68**	**0.57**	**0.61**	**0.79**

MPS-VAS framework for visual active search in real-world scenarios where the search tasks differ from the ones used during policy training.

Table 16: **ANT** comparisons when trained with *LV* as target on DOTA in single-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
RS	1.45	3.17	4.30	1.79	3.50	5.10	2.35	4.34	6.76
E2EVAS [6]	2.69	4.50	5.88	4.63	6.79	8.07	4.22	6.92	9.06
OnlineTTA[6]	2.70	4.52	5.89	4.63	6.80	8.07	4.22	6.93	9.08
PSVAS	**3.19**	**4.83**	**6.34**	**4.69**	**6.94**	**8.12**	**4.95**	**7.56**	**9.51**
MPS-VAS	**3.42**	**5.19**	**6.73**	**4.80**	**7.08**	**8.23**	**5.02**	**8.04**	**9.91**
	Test with Helicopter as Target			**Test with Plane as Target**			**Test with RB as Target**		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
RS	0.60	1.27	1.96	2.33	4.34	6.62	0.64	1.06	1.80
E2EVAS [6]	1.00	2.07	2.66	4.57	7.23	9.14	1.56	2.28	2.72
OnlineTTA[6]	1.00	2.07	2.68	4.57	7.25	9.16	1.56	2.28	2.73
PSVAS	**1.53**	**2.33**	**2.84**	**5.09**	**7.64**	**9.41**	**1.87**	**2.34**	**2.76**
MPS-VAS	**1.80**	**2.60**	**3.03**	**5.17**	**7.83**	**10.02**	**1.96**	**2.76**	**3.19**

Table 17: **ANT** comparisons when trained with *LV* as target on DOTA in multi-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
MPS-VAS-TOPK	3.33	5.14	6.70	4.64	6.83	7.79	4.96	7.91	9.75
MPS-VAS-MQ	**3.38**	**5.17**	**6.71**	**4.65**	**6.92**	**8.00**	**4.99**	**7.98**	**9.83**
	Test with Helicopter as Target			Test with Plane as Target			Test with RB as Target		
Method	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$	$\mathcal{C}=25$	$\mathcal{C}=50$	$\mathcal{C}=75$
MPS-VAS-TOPK	1.34	2.42	2.88	5.08	7.63	9.66	1.76	2.68	3.02
MPS-VAS-MQ	**1.37**	**2.43**	**2.91**	**5.15**	**7.75**	**9.95**	**1.82**	**2.72**	**3.11**

Table 18: **ANT** comparisons when trained with *LV* as target on DOTA in single-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
RS	2.92	3.34	3.99	3.44	4.08	5.19	4.17	5.04	5.92
E2EVAS [6]	3.34	4.15	4.77	5.14	6.05	7.00	5.38	6.51	7.54
OnlineTTA[6]	3.36	4.15	4.79	5.14	6.06	7.01	5.40	6.52	7.55
PSVAS	**3.48**	**4.37**	**5.15**	**5.23**	**6.08**	**7.12**	**5.57**	**6.69**	**7.78**
MPS-VAS	**3.85**	**4.69**	**5.38**	**5.25**	**6.11**	**7.14**	**5.71**	**6.95**	**8.15**
	Test with Helicopter as Target			Test with Plane as Target			Test with RB as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
RS	1.03	1.52	1.77	4.05	5.11	6.12	1.25	1.54	1.91
E2EVAS [6]	1.50	1.87	2.13	5.47	6.59	7.65	1.87	2.17	2.47
OnlineTTA[6]	1.50	1.88	2.16	5.47	6.61	7.68	todo	todo	todo
PSVAS	**1.77**	**2.23**	**2.50**	**5.54**	**6.65**	**7.66**	**2.03**	**2.32**	**2.65**
MPS-VAS	**2.10**	**2.57**	**2.77**	**5.73**	**6.87**	**7.90**	**2.12**	**2.66**	**2.99**

Table 19: **ANT** comparisons when trained with *LV* as target on DOTA in multi-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
MPS-VAS-TOPK	3.84	4.64	5.28	5.14	6.01	6.51	5.65	6.84	7.93
MPS-VAS-MQ	**3.81**	**4.64**	**5.35**	**5.22**	**6.05**	**6.68**	**5.66**	**6.89**	**8.04**
	Test with Helicopter as Target			Test with Plane as Target			Test with RB as Target		
Method	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$	$\mathcal{C}=12$	$\mathcal{C}=15$	$\mathcal{C}=18$
MPS-VAS-TOPK	1.39	1.91	2.27	5.64	6.79	7.71	2.01	2.43	2.68
MPS-VAS-MQ	**1.43**	**1.96**	**2.33**	**5.65**	**6.83**	**7.80**	**2.08**	**2.49**	**2.81**

D Effect of Inference Time Adaptation of Task Specific Prediction Module on Search Performance

D.1 Effect on PSVAS Framework

First, we analyze the impact of inference time adaptation of task specific prediction module on PSVAS framework across different target settings. To this end, we first train a policy using our proposed PSVAS approach and then during inference we freeze the task specific prediction module along with task agnostic search module unlike PSVAS approach. We call the resulting policy as PSVAS-F. In Table 20, we compare the search performance of PSVAS and PSVAS-F with number of grid cell $N = 36$ across different target settings. In Table 21, we present similar results with number of grid cell $N = 49$. We observe a significant improvement in performance of PSVAS compared to PSVAS-F across different target settings, justifying the importance of inference time adaptation of task specific prediction module after every query.

Table 20: **ANT** comparisons when trained with *LV* as target on DOTA in single-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$
PSVAS-F	2.77	4.55	5.99	4.61	6.77	8.09	4.26	6.87	9.05
PSVAS	**3.19**	**4.83**	**6.34**	**4.69**	**6.94**	**8.12**	**4.95**	**7.56**	**9.51**
	Test with Helicopter as Target			Test with Plane as Target			Test with RB as Target		
Method	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$
PSVAS-F	1.02	2.03	2.64	4.62	7.26	9.16	1.57	2.29	2.72
PSVAS	**1.53**	**2.33**	**2.84**	**5.09**	**7.64**	**9.41**	**1.87**	**2.34**	**2.76**

Table 21: **ANT** comparisons when trained with *small car* as target on xView in single-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$
PSVAS-F	0.55	0.86	1.24	0.66	1.12	1.34	5.88	9.45	12.23
PSVAS	**0.87**	**1.08**	**1.28**	**0.93**	**1.23**	**1.66**	**6.81**	**10.53**	**13.44**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$	$\mathcal{C} = 25$	$\mathcal{C} = 50$	$\mathcal{C} = 75$
PSVAS-F	1.45	2.30	3.01	4.84	7.56	9.65	0.82	1.20	1.46
PSVAS	**1.62**	**2.49**	**3.14**	**5.51**	**8.33**	**10.52**	**0.91**	**1.22**	**1.47**

In Figure 4, the distinct exploration strategy behaviors of PSVAS and PSVAS-F are depicted when both policies are trained with a *large vehicle* as the target and tested with a *ship* as the target. Out of a total of 15 queries, PSVAS-F achieves 6 successful searches, while PSVAS achieves 8 successful searches. Figure 5 illustrates the contrasting exploration strategy behaviors between PSVAS and

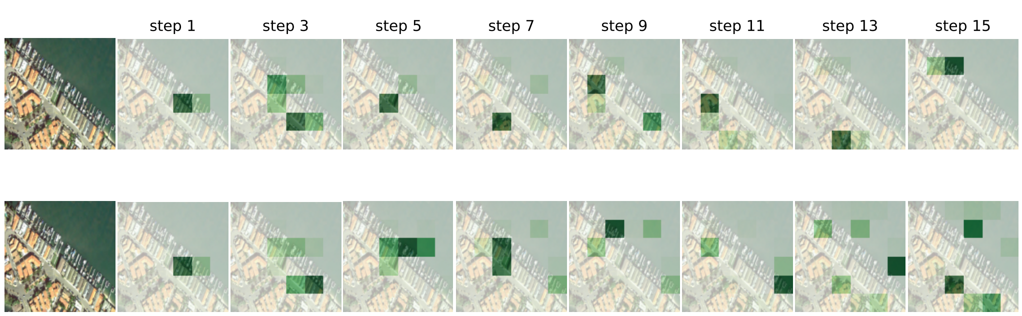

Figure 4: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using PSVAS-F (top row), PSVAS (bottom row).

PSVAS-F in the case when both the policies are trained with *large vehicle* as the target and test with *plane* as the target. We observe PSVAS-F yields 9 successful searches, while PSVAS yields 12 successful search out of 15 total query.

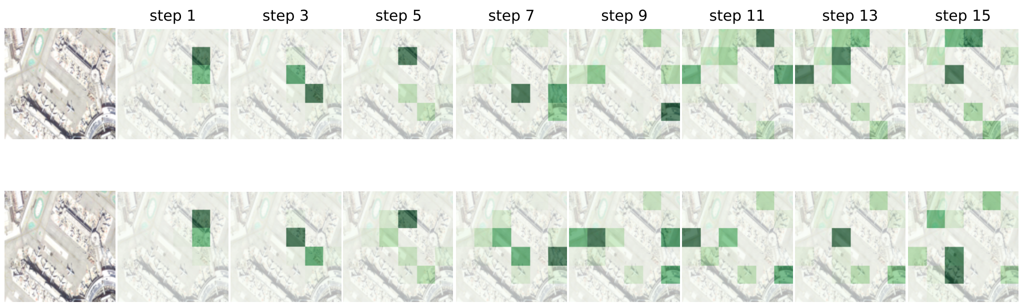

Figure 5: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using PSVAS-F (top row), PSVAS (bottom row).

Figure 6 illustrates the contrasting exploration strategy behaviors between PSVAS and PSVAS-F in the case when both the policies are trained with *large vehicle* as the target and test with *roundabout* as the target. We observe PSVAS-F yields 5 successful searches, while PSVAS yields 7 successful search out of 15 total query.

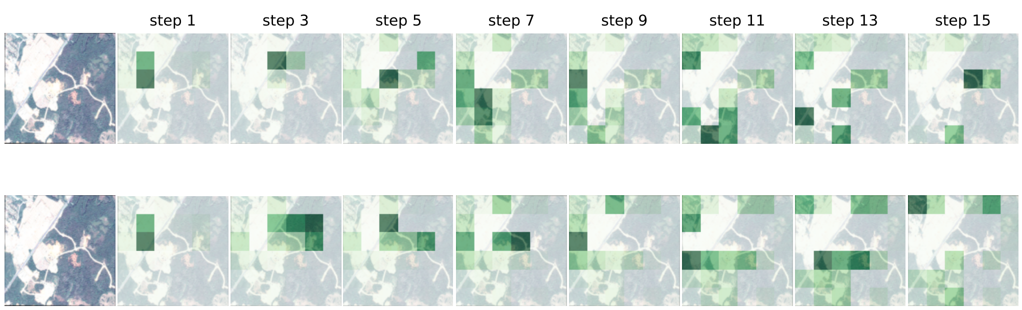

Figure 6: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using PSVAS-F (top row), PSVAS (bottom row).

D.2 Effect on MPS-VAS Framework

Next, we examine the influence of inference time adaptation of the task-specific prediction module on the MPS-VAS framework across various target settings. For this purpose, we train a policy using our proposed MPS-VAS approach. But during inference, we freeze both the task-specific prediction module and the task-agnostic search module, which differs from the standard MPS-VAS approach. We refer the resulting policy as MPS-VAS-F. Table 23 presents a comparison of the search performance between MPS-VAS and MPS-VAS-F, considering a grid cell count of $N = 36$, across various target settings. Similarly, in Table 22, we provide corresponding results with a grid cell count of $N = 49$. Across various target settings, we observe a notable enhancement in the performance of MPS-VAS compared to MPS-VAS-F. This finding underscores the significance of adapting the task-specific prediction module during inference after each query, validating its importance on adaptive visual active search. Following Figures demonstrate the divergent exploration strategy behaviors exhibited by MPS-VAS and MPS-VAS-F.

Figure 7 illustrates the contrasting exploration strategy behaviors of MPS-VAS and MPS-VAS-F when both policies are trained with a *large vehicle* as the target and tested with a *plane* as the target.

Table 22: **ANT** comparisons when trained with *small car* as target on xView in single-query setting.

	Test with Helicopter as Target			Test with SB as Target			Test with Building as Target		
Method	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75
MPS-VAS-F	0.54	0.89	1.22	0.64	1.14	1.37	5.97	9.31	12.04
MPS-VAS	**0.92**	**1.13**	**1.38**	**1.07**	**1.67**	**2.10**	**6.83**	**10.59**	**13.64**
	Test with CC as Target			Test with SC as Target			Test with Helipad as Target		
Method	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75
E2EVAS [6]	1.37	2.33	3.05	4.82	7.46	9.56	0.82	1.24	1.41
MPS-VAS	**1.74**	**2.64**	**3.47**	**5.55**	**8.40**	**10.69**	**0.96**	**1.30**	**1.63**

Table 23: **ANT** comparisons when trained with *LV* as target on DOTA in single-query setting.

	Test with Ship as Target			Test with LV as Target			Test with Harbor as Target		
Method	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75
E2EVAS [6]	2.69	4.50	5.88	4.63	6.79	8.07	4.22	6.92	9.06
MPS-VAS	**3.42**	**5.19**	**6.73**	**4.80**	**7.08**	**8.23**	**5.02**	**8.04**	**9.91**
	Test with Helicopter as Target			Test with Plane as Target			Test with RB as Target		
Method	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75	$\mathcal{C}$ = 25	$\mathcal{C}$ = 50	$\mathcal{C}$ = 75
E2EVAS [6]	1.00	2.07	2.66	4.57	7.23	9.14	1.56	2.28	2.72
MPS-VAS	**1.80**	**2.60**	**3.03**	**5.17**	**7.83**	**10.02**	**1.96**	**2.76**	**3.19**

Among a total of 15 queries, MPS-VAS-F achieves 2 successful searches, while MPS-VAS achieves 4 successful searches. In Figure 8, the distinct exploration strategy behaviors of MPS-VAS and

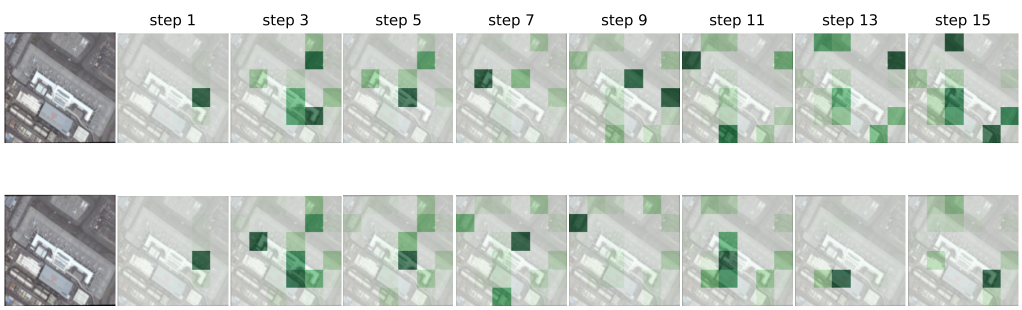

Figure 7: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using MPS-VAS-F (top row), MPS-VAS (bottom row).

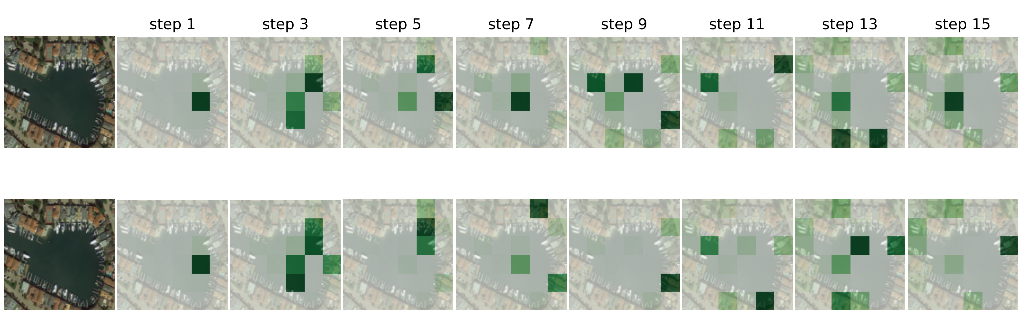

Figure 8: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using MPS-VAS-F (top row), MPS-VAS (bottom row).

MPS-VAS-F are depicted when both policies are trained with a *large vehicle* as the target and tested

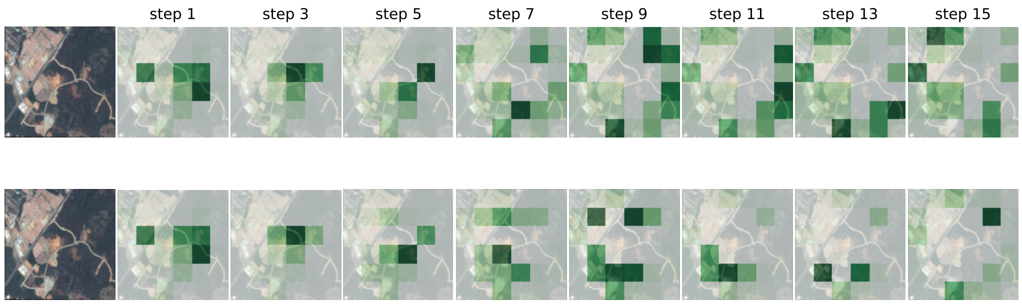

Figure 9: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using MPS-VAS-F (top row), MPS-VAS (bottom row).

with a *ship* as the target. Out of a total of 15 queries, MPS-VAS-F achieves 7 successful searches, while MPS-VAS achieves 9 successful searches. Figure 9 showcases the contrasting exploration strategy behaviors of MPS-VAS and MPS-VAS-F when both policies are trained with a *large vehicle* as the target and tested with a *roundabout* as the target. Among a total of 15 queries, MPS-VAS-F achieves 6 successful searches, while MPS-VAS achieves 8 successful searches.

E More Visualizations of Comparative Exploration Strategies of Different Approaches

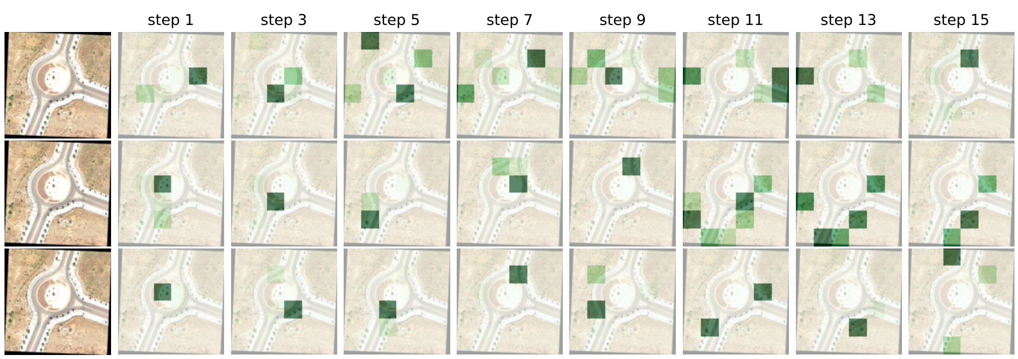

Figure 10: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using E2EVAS (top row), PSVAS (middle row), and MPS-VAS (bottom row). Note that during the training phase, all these policies are trained with *large vehicle* as the target, while evaluation is conducted using *roundabout* as the target.

The showcased visualizations (10, 11, 12, 13, 14) in all these examples demonstrate the superiority of our PSVAS and MPS-VAS framework compared to the E2EVAS baseline, especially in scenarios where search tasks vary from those employed in policy training.

F Analyzing Search Performance Across Multiple Trials

Here, we compare the search performance of E2EVAS, PSVAS, and MPS-VAS across multiple trials. In Figure 15, we present the results when the polices are trained with small car as the target and evaluate the performance under Manhattan distance based query cost $\mathcal{C} = 25$ with the following target classes: *Small Car (SC)*, *Helicopter*, *Sail Boat (SB)*, *Construction Cite (CC)*, *Building*, and *Helipad*. In figure 16, we present similar results with Manhattan distance based query cost budget $\mathcal{C} = 50$. In figure 17, we also present similar results with Manhattan distance based query cost budget $\mathcal{C} = 75$.

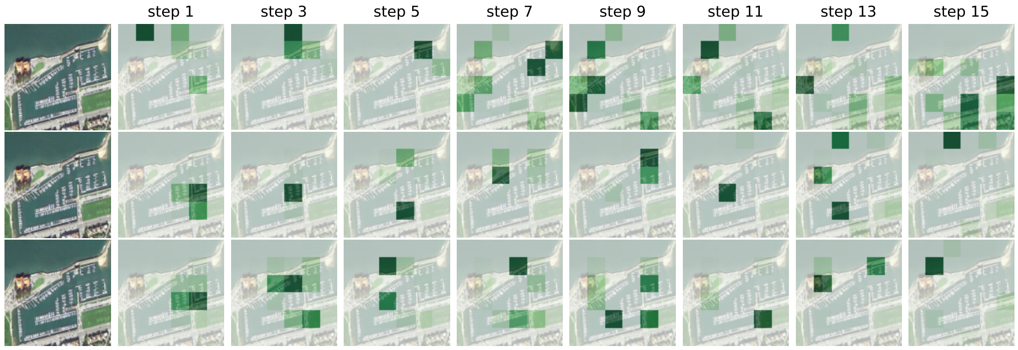

Figure 11: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using E2EVAS (top row), PSVAS (middle row), and MPS-VAS (bottom row). Note that during the training phase, all these policies are trained with *large vehicle* as the target, while evaluation is conducted using *ship* as the target.

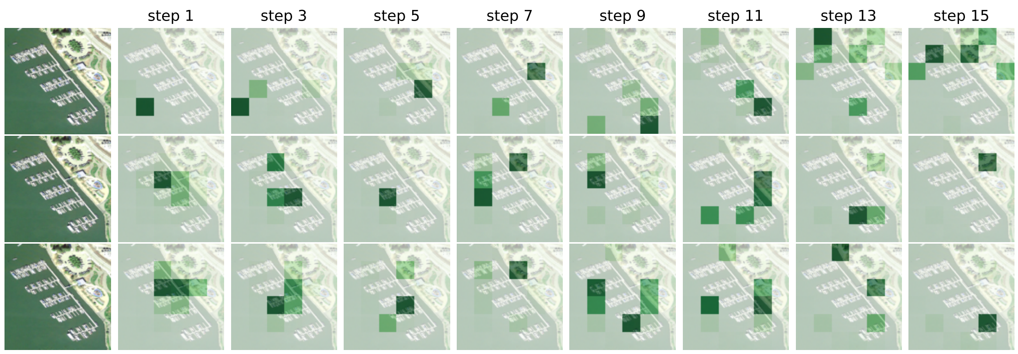

Figure 12: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using E2EVAS (top row), PSVAS (middle row), and MPS-VAS (bottom row). Note that during the training phase, all these policies are trained with *large vehicle* as the target, while evaluation is conducted using *ship* as the target.

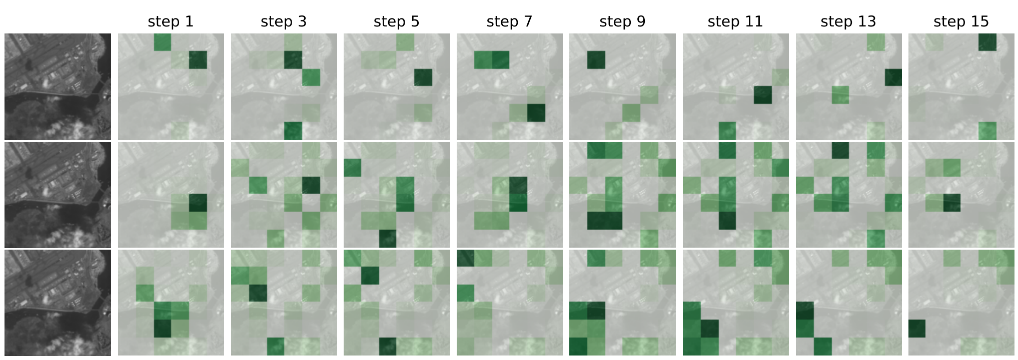

Figure 13: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using E2EVAS (top row), PSVAS (middle row), and MPS-VAS (bottom row). Note that during the training phase, all these policies are trained with *large vehicle* as the target, while evaluation is conducted using *plane* as the target.

In Figure 18, we present the results when the polices are trained with *large vehicle* as the target and evaluate the performance under Manhattan distance based query cost $\mathcal{C}$ = 25 with the following target classes: *Ship*, *large vehicle* (LV), *Harbor*, *Helicopter*, *Plane*, and *Roundabout*. In figure 19, we

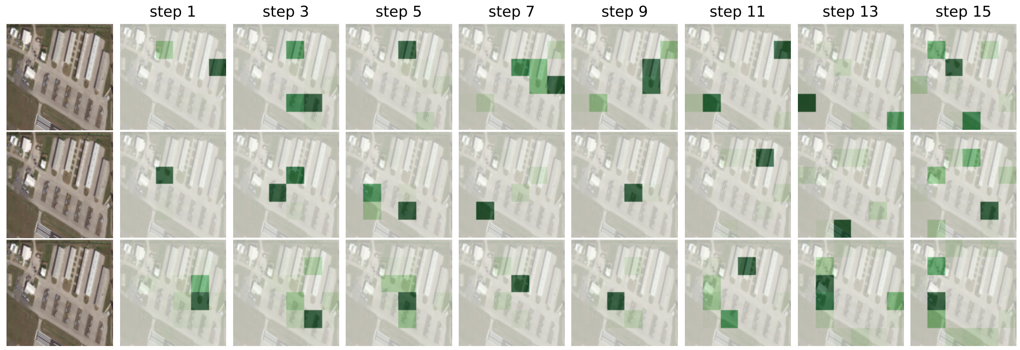

Figure 14: Query sequences, and corresponding heat maps (darker indicates higher probability), obtained using E2EVAS (top row), PSVAS (middle row), and MPS-VAS (bottom row). Note that during the training phase, all these policies are trained with *large vehicle* as the target, while evaluation is conducted using *plane* as the target.

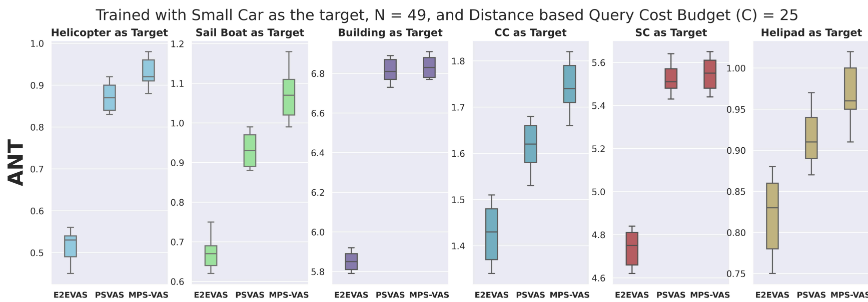

Figure 15: Comparative Search Performance of E2EVAS, PSVAS, MPS-VAS under Distance Based Query Cost ($\mathcal{C}$ = 25).

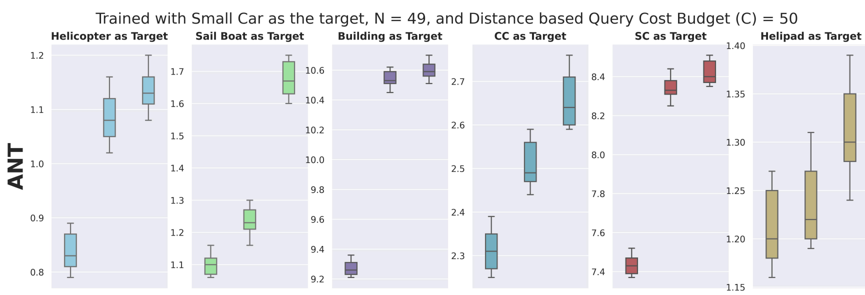

Figure 16: Comparative Search Performance of E2EVAS, PSVAS, MPS-VAS under Distance Based Query Cost ($\mathcal{C}$ = 50).

present similar results with Manhattan distance based query cost budget $\mathcal{C}$ = 50. In figure 20, we also present similar results with Manhattan distance based query cost budget $\mathcal{C}$ = 75.

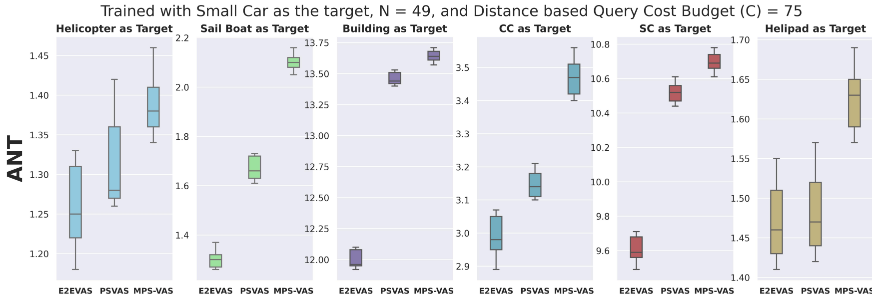

Figure 17: Comparative Search Performance of E2EVAS, PSVAS, MPS-VAS under Distance Based Query Cost ($\mathcal{C}$ = 75).

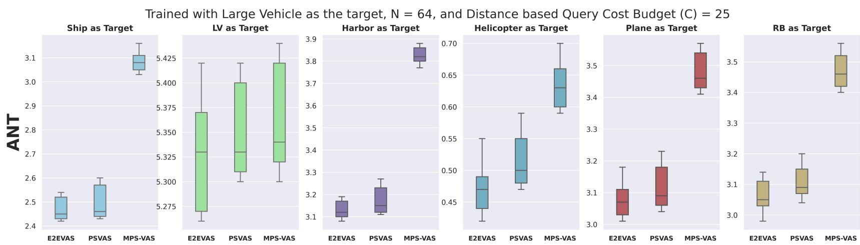

Figure 18: Comparative Search Performance of E2EVAS, PSVAS, MPS-VAS under Distance Based Query Cost ($\mathcal{C}$ = 25).

Figure 19: Comparative Search Performance of E2EVAS, PSVAS, MPS-VAS under Distance Based Query Cost ($\mathcal{C}$ = 50).

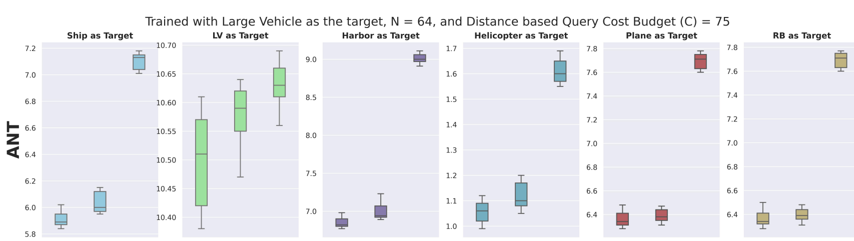

Figure 20: Comparative Search Performance of E2EVAS, PSVAS, MPS-VAS under Distance Based Query Cost ($\mathcal{C}$ = 75).