Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion

Subhabrata Choudhury*
subha@robots.ox.ac.uk
Laurynas Karazija*
laurynas@robots.ox.ac.uk
Iro Laina
iro@robots.ox.ac.uk
Andrea Vedaldi
vedaldi@robots.ox.ac.uk
Christian Rupprecht
chrisr@robots.ox.ac.uk

Visual Geometry Group
University of Oxford
Oxford, UK

Supplementary Material

In this supplementary material, we provide further details on our training parameters in Appendix A. Appendix B contains the closed form solution of the fitting of the flow model θ. Expanded experiments and ablations are found in Appendix C. Finally, more qualitative results are presented in Appendix D. See the project page, https://www.robots.ox.ac.uk/~vgg/research/gwm, for additional visualizations, code and models.

A Experimental Setup

Network. We use MaskFormer [5] as our segmentation network[1], and use only the segmentation head. As MaskFormer predicts masks at 4 times lower resolution than input, we modify the `PixelDecoder` by appending [*Conv*(3), *UpsampleNN(2)*, *Conv*(1)]×2 to its output layers to bring the masks back up to the input resolution.

For the backbone and appearance features V, we leverage a ViT-8 transformer, pretrained on ImageNet [13] in a self-supervised manner using DINO [3] to avoid any external sources of supervision. For the hierarchical backbone features to decoder we use the key feature outputs from layers 6, 8, 10, 12.

The input RGB images are interpolated (bi-cubic) to 128×224 resolution for input to the network. We interpolate (nearest neighbor) the optical flow to 480×854 for the loss. Output

*Authors contributed equally.

[1] Implementation from https://github.com/facebookresearch/MaskFormer.

segmentation logits are up-sampled using bi-linear interpolation to the flow resolution for training and again to annotation resolution for evaluation.

Training Hyperparameters. The networks are optimised using AdamW [11], with learning rate of 1.5×10^{-4}, a schedule of linear warm-up from 1.0×10^{-6} to 1.5×10^{-4} over 1.5k iteration and polynomial decay afterwards. We use batch size of 8 and train for 15k iterations. We additionally employ gradient clipping when the 2-norm exceeds 0.01 for stability. The loss multiplier is 0.03.

UNet. For experiments using U-Net[2], we use the standard 4-layer version. The batch-size is increased to 16 and learning rate to 7.0×10^{-4}. We also clip the gradients only when 2-norm exceeds 5.0. All other settings, including optimizer and learning rate schedules, are kept the same. U-Net is not pre-trained and trained from scratch.

Optical Flow. Our method derives its learning signal from optical flow estimated using off-the-shelf frozen networks. We estimate optical flow for all frames on DAVIS, STv2, and FBMS following the practice of MotionGrouping [27]. We employ RAFT [13] (supervised) using the original resolution for our main experiments, and gaps between frames of $\{-2,-1,1,2\}$ for DAVIS and STv2, and $\{-6,-3,3,6\}$ on FBMS. When multiple flows are associated with a single frame (multiple gaps), we sample one at random for each iteration.

B Quadratic Flow Model: Closed Form Solution

Consider one of K regions m and define $w_u \propto P(m_u = k|I,\Phi)$ the posterior probability for that region, normalized so that $\sum_{u\in\Omega} w_u = 1$ (the scaling factor does not matter for the purpose of finding the minimizer). We can obtain the minimizer (A^*, b^*) and minimum of the energy

$$E(A,b) = \sum_{u\in\Omega} w_u \|F_u - Au - b\|^2 \tag{1}$$

as follows. Defining

$$\bar{u} := \begin{bmatrix} u \\ 1 \end{bmatrix}, \quad M := \begin{bmatrix} A & b \end{bmatrix} \in \mathbb{R}^{2\times 6}$$

allows rewriting the energy as

$$E(M) = \sum_{u\in\Omega} w_u \|F_u - M\bar{u}\|^2 = \operatorname{tr}\left(\Lambda_{FF} - M\Lambda_{\bar{\Omega}F} - \Lambda_{F\bar{\Omega}}M^\top + M\Lambda_{\bar{\Omega}\bar{\Omega}}M^\top\right),$$

where

$$\Lambda_{FF} = \sum_{u\in\Omega} w_u F_u F_u^\top, \;\; \Lambda_{F\bar{\Omega}} = \sum_{u\in\Omega} w_u F_u \bar{u}^\top, \;\; \Lambda_{\bar{\Omega}F} = \Lambda_{F\bar{\Omega}}^\top, \;\; \Lambda_{\bar{\Omega}\bar{\Omega}} = \sum_{u\in\Omega} w_u \bar{u}\bar{u}^\top.$$

are the (uncentered) second moment matrices of the flow F_u and homogeneous coordinate vectors $\bar{u}$. By inspection of the trace term, the gradient of the energy is given by:

$$\frac{dE(M)}{dM} = 2\left(\Lambda_{F\bar{\Omega}} - M\Lambda_{\bar{\Omega}\bar{\Omega}}\right)$$

Hence, the optimal regression matrix M^* and corresponding energy value are

$$M^* = \Lambda_{F\bar{\Omega}}\Lambda_{\bar{\Omega}\bar{\Omega}}^{-1}, \quad E(M^*) = \operatorname{tr}\left(\Lambda_{FF} - M^*\Lambda_{\Omega\bar{F}}\right).$$

[2]Implementation from https://github.com/milesial/Pytorch-UNet.

Somewhat more intuitive results can be obtained by centering the moments and resolving for A and b instead of M. Specifically, define:

$$\mu_\Omega := \sum_{u\in\Omega} w_u u, \quad \mu_F := \sum_{u\in\Omega} w_u F_u.$$

The covariance matrices of the vectors are:

$$\Sigma_{FF} = \sum_{u\in\Omega} w_u (F_u-\mu_F)(F_u-\mu_F)^\top, \quad \Sigma_{F\Omega} = \sum_{u\in\Omega} w_u (F_u-\mu_F)(u-\mu_\Omega)^\top,$$
$$\Sigma_{\Omega F} = \Lambda_{F\Omega}^\top, \quad \Sigma_{\Omega\Omega} = \sum_{u\in\Omega} w_u (u-\mu_\Omega)(u-\mu_\Omega)^\top.$$

It is easy to check that

$$\Lambda_{FF} = \Sigma_{FF} + \mu_F\mu_F^\top, \quad \Lambda_{F\bar\Omega} = \begin{bmatrix}\Sigma_{F\Omega}+\mu_F\mu_\Omega^\top & \mu_F\end{bmatrix}, \quad \Lambda_{\bar\Omega\bar\Omega} = \begin{bmatrix}\Sigma_{\Omega\Omega}+\mu_\Omega\mu_\Omega^\top & \mu_\Omega \\ \mu_\Omega^\top & 1\end{bmatrix}.$$

From this:

$$\begin{aligned} M^* = \Lambda_{F\bar\Omega}\Lambda_{\bar\Omega\bar\Omega}^{-1} &= \begin{bmatrix}\Sigma_{F\Omega}+\mu_F\mu_\Omega^\top & \mu_F\end{bmatrix}\begin{bmatrix}\Sigma_{\Omega\Omega}+\mu_\Omega\mu_\Omega^\top & \mu_\Omega \\ \mu_\Omega^\top & 1\end{bmatrix}^{-1} \\ &= \begin{bmatrix}\Sigma_{F\Omega}+\mu_F\mu_\Omega^\top & \mu_F\end{bmatrix}\begin{bmatrix}\Sigma_{\Omega\Omega}^{-1} & -\Sigma_{\Omega\Omega}^{-1}\mu_\Omega \\ -\mu_\Omega^\top\Sigma_{\Omega\Omega}^{-1} & 1+\mu_\Omega^\top\Sigma_{\Omega\Omega}^{-1}\mu_\Omega\end{bmatrix} \\ &= \begin{bmatrix}\Sigma_{F\Omega}\Sigma_{\Omega\Omega}^{-1} & \mu_F - \Sigma_{F\Omega}\Sigma_{\Omega\Omega}^{-1}\,\mu_\Omega\end{bmatrix} = \begin{bmatrix}A^* & b^*\end{bmatrix}. \end{aligned}$$

Hence, the optimal regression coefficients and energy value are also given by:

$$A^* = \Sigma_{F\Omega}\Sigma_{\Omega\Omega}^{-1}, \quad b^* = \mu_F - A^*\mu_\Omega.$$

C Further Experiments

C.1 Generalization in Unsupervised Video Segmentation

We also test our model in a video *generalization* setting. In contrast to the protocol of [27, 29], where evaluation set is observed together with training to infer masks jointly[3], here we train only on frames from the training set. We report performance on unseen videos. In this case, our method independently segments a collection of frames from a new video, with no way to incorporate motion information.

To "observe" motion on *unseen* inputs, we also report results after taking 20 test-time adaptation steps (using our unsupervised loss) for each evaluation sequence in isolation (c.f. AMD [10] takes 100 test-time adaptations steps). That is after training, we follow our training setup (optimizer, rate, batch size) and feed frames from the evaluation video and corresponding optical flow, calculate loss and take gradient steps. Despite other methods using much larger training sets, our approach shows better performance (Table 1).

	Model	Flow	**DAVIS** ($\mathcal{J}\uparrow$)	**FBMS** ($\mathcal{J}\uparrow$)
[10]	AMD (100 steps)	✗	57.8	47.5
	Ours (Zero shot)	ARFlow	62.5	65.4
	Ours (20 steps)	ARFlow	65.2	67.6
[12]	EM	RAFT	69.3	57.8
	Ours (Zero shot)	RAFT	66.8	73.2
	Ours (20 steps)	RAFT	76.3	77.1

Table 1: **Generalization performance on unseen *videos*.** Few unsupervised methods operate in this setting. AMD trains on YT-VOS, followed by 100 test-time adaptation steps, while EM trains on FlyingThings3D using flow as input. We use (fully unsupervised) ARFlow for fair comparison with AMD. Our method shows better performance after observing motion. (Test-time adaptation uses the training loss. No GT is involved at any point.)

Backbone model	Backbone pretraining	Sup.	**DAVIS** $\mathcal{J}\uparrow$	**STv2** $\mathcal{J}\uparrow$	**FBMS** $\mathcal{J}\uparrow$
ViT-8	ImageNet DINO	✗	79.5	78.3	77.4
UNet	None	✗	78.3	76.8	72.0
SWIN-tiny	ImageNet MOBY	✗	78.3	77.4	74.6
SWIN-tiny	ImageNet CLS	✓	78.9	77.7	75.5
SWIN-tiny	None	✗	78.3	75.2	68.8
Resnet-50	ImageNet CLS	✓	77.5	75.8	72.9

Table 2: **Effect of Pretraining/Backbone.** Our method with MaskFormer benefits from pretraining, with slight improvement offered by supervised (*CLS*) over unsupervised (*MOBY*) pretraining (usng SWIN transformer). Comparable results can be obtained with training from scratch. Best results are obtained using DINO features.

C.2 Ablation Studies

Pretraining. Compared to recent methods for video segmentation [14, 22], one of the benefits of our formulation is that we can leverage unsupervised pretraining for the segmentation network (*e.g.*, for the ViT backbone of MarkFormer). This enables our method to be trained in only 15k iterations. Here, we investigate the importance of the backbone. To this end we replace ViT with Swin-tiny pretrained using MOBY (self-supervised) in Table 2. The performance differences are small.

Additionally, we investigate the effect of other pretraining strategies on the performance. Switching to a model pretrained on ImageNet with image-level supervision (*i.e.* a classification task) only slightly improves performance showing that the method does not need to rely on supervised pre-training. Finally, we train the model using same settings for 20k iterations from scratch, without any pre-training. This results in comparable performance on DAVIS but reduced performance on the smaller datasets. Comparing backbones without pre-training, UNet gives better results than SWIN-tiny, likely due to smaller networks being easier to train on small datasets.

Feature Clustering without Motion. To demonstrate the potential of using motion for discovering objects, in Table 3, we compare to additional baselines that only rely on clustering visual features. Spectral feature clustering with $K = 2$ (based on [17]), on the same visual

[3] Note, no annotations are observed at any point.

Model	K	Merge	DAVIS $\mathcal{J}\uparrow$	STv2 $\mathcal{J}\uparrow$	FBMS $\mathcal{J}\uparrow$
Ours	$K=4$	✓	**79.5**	**78.3**	**77.4**
Spectral clustering	$K=2$	✗	15.79	14.89	27.45
K-Means	$K=4$	✓	41.79	34.84	48.80
K-Means	$K=2$	✗	20.24	21.14	38.25

Table 3: **Feature Clustering without Motion.** We experiment with offline clustering of DINO features to assess the importance of our motion-based formulation. Simply clustering DINO features using K-Means or spectral clustering [47] into 2 clusters performs worse. Over-clustering and merging using our cluster-merging approach performs better but still fails to reach our performance.

Opt. Flow		Sup.	DAVIS ($\mathcal{J}\uparrow$)
[9]	ARFlow	✗	66.9
[47]	PWCNet	✓	74.9
[28]	RAFT	✓	79.5

Table 4: **Choice of Optical Flow Method.** Measuring the influence of the method to extract optical flow.

Method		DAVIS ($\mathcal{J}\uparrow$)
[27]	MG	53.2
[10]	AMD	57.8
	Ours	**66.9**

Table 5: **Fully Unsupervised Video Object Segmentation.** Comparison to the state of the art in unsupervised VOS without reliance on *any* supervision

features we use to merge segments (*i.e.*, DINO) after over-clustering, shows (somewhat unsurprisingly) that learning from motion is important for motion segmentation. Similarly, K-means ($K=2$) on the same features also falls behind our method. Yet, we show that K-means also benefits from over-clustering ($K=4$) and then merging.

Flow Estimation. Finally, our method relies on optical flow estimated by frozen, off-the-shelf networks. So far we have been using RAFT [28], as such optical flow network was adopted in our baselines. In Table 4, we also consider PWCNet [47] and fully-unsupervised ARFlow [9]. We observe that the performance of the flow estimator has an impact on the final performance of our method. Finally, we compare our *fully* unsupervised model (which uses self-supervised pretraining and flow) to fully unsupervised state-of-the-art methods. Appearance-Motion Decomposition (AMD) [10] works end-to-end and directly extracts motion features from pairs of images with a PWCNet-like architecture, while MotionGrouping (MG) [27] and our method use ARFlow [9] for optical flow estimation. In Table 5 we show that our method achieves a significant improvement over previous approaches.

D Additional Results and Discussion

We provide a further breakdown of our results in Tables 7 to 9, reporting per sequence evaluation results on the video segmentation tasks.

Video object segmentation and egomotion. We note that some sequences have pronounced egomotion (*e.g.*, camera shaking in `libby` of DAVIS or inside a moving car in `camel01`

	CUB			DUTS			ECSSD			OMRON		
	Acc	$\mathcal{J}\uparrow$	$maxF_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$	Acc	$\mathcal{J}\uparrow$	$F_\beta\uparrow$
[WNet†]	–	24.8	–	–	–	–	–	–	–	–	–	–
[IIC-seg]	–	36.5	–	–	–	–	–	–	–	–	–	–
[PertGAN]	–	38.0	–	–	–	–	–	–	–	–	–	–
[ReDO]	84.5	42.6	–	–	–	–	–	–	–	–	–	–
[UISB]	–	44.2	–	–	–	–	–	–	–	–	–	–
[OneGAN]	–	55.5	–	–	–	–	–	–	–	–	–	–
[DRC]	–	56.4	–	–	–	–	–	–	–	–	–	–
[GANSeg]	–	62.9	–	–	–	–	–	–	–	–	–	–
[Voynov *et al.*]	94.0	71.0	80.7	88.1	51.1	60.0	90.6	68.4	79.0	86.0	46.4	53.3
[AMD]	–	–	–	–	–	60.2	–	–	–	–	–	–
[Kyriazi *et al.*]	92.1	66.4	78.3	89.3	52.8	61.4	91.5	71.3	80.6	88.3	50.9	58.3
[Kyriazi *et al.*]	–	76.9	–	–	51.4	–	–	73.3	–	56.7	–	
[DySta B†]	–	–	–	–	–	–	–	–	88.1	–	–	73.9
[TokenCut]	–	–	–	90.3	57.6	–	91.8	71.2	–	88.0	53.3	–
[SelfMask]	–	–	–	92.3	62.6	–	94.4	78.1	–	90.1	58.2	–
Ours	93.5	64.6	80.9	91.5	49.2	65.6	88.5	56.1	74.3	89.3	41.31	56.3

Table 6: **Expanded unsupervised object segmentation** benchmark CUB and three saliency detection benchmarks: DUTS, ECSSD, and DUT-OMRON (*OMRON*). † DySta B uses CRF post-processing, supervised pre-training, and self-training on each dataset.

of FBMS). Our model performs well on these sequences, demonstrating that it can handle egomotion. When *only* the camera is moving, the resulting optical flow would still highlight objects due to parallax. This provides a learning signal, however, it would likely be weaker for objects farther away from the camera. As our method works on a per-frame basis and does not *require* flow during inference, this should not have an impact at test time. However, fine-tuning on scenes with only egomotion (see Appendix C.1 for experiments investigating test-time adaptation) and only small or far away objects, might lead to the model learning to ignore them.

Image segmentation. For unsupervised image segmentation, we show some additional qualitative results for CUB in Fig. 1, DUT-OMRON in Fig. 2, DUTS in Fig. 3, and ECSSD in Fig. 4. Our model, trained on a combined dataset of DAVIS, FBMS and STv2, is robust enough to handle a wide array of classes from the above datasets in varying context. Our model can segment both stationary and non-stationary objects and works well when multiple objects are in the foreground. In Fig. 5, we show a few failure cases for all datasets, where the model struggles mostly with ambiguous foreground objects and, in particular, with close-ups of stationary objects, *e.g.* signs (ECSSD) and buildings (DUT-OMRON). The model also has issues with boundaries for many objects, *i.e.* the foreground objects are correctly identified but the model fails to fully segment them. For example, in DUTS, the snake in the first image has a well segmented head, however, the model does not segment its body accurately.

Sequence	*w/o CRF*			*w/ CRF*		
	$\mathcal{J}$(M)	$\mathcal{J}$(R)	$\mathcal{J}$(D)	$\mathcal{J}$(M)	$\mathcal{J}$(R)	$\mathcal{J}$(D)
blackswan	67.0	100.0	-0.8	67.4	100.0	1.1
bmx-trees	58.2	76.9	19.9	59.8	76.9	17.5
breakdance	86.2	100.0	4.9	87.4	100.0	5.2
camel	89.4	100.0	5.7	90.6	100.0	5.5
car-roundabout	81.4	90.4	26.7	81.2	90.4	25.8
car-shadow	84.3	100.0	9.0	83.9	100.0	8.0
cows	90.4	100.0	3.4	91.3	100.0	3.2
dance-twirl	87.4	100.0	-7.1	88.8	100.0	-6.2
dog	92.9	100.0	-1.7	93.9	100.0	-1.6
drift-chicane	78.6	98.0	2.2	82.0	100.0	2.6
drift-straight	80.6	100.0	7.2	82.1	100.0	8.2
goat	78.6	100.0	1.7	75.8	100.0	4.5
horsejump-high	84.9	100.0	6.4	88.0	100.0	4.6
kite-surf	64.4	97.9	4.5	67.5	97.9	3.1
libby	82.9	100.0	8.6	84.5	100.0	8.6
motocross-jump	74.1	78.9	4.1	75.1	81.6	4.1
paragliding-launch	62.2	65.4	33.5	64.1	66.7	35.8
parkour	86.1	100.0	-4.5	88.1	100.0	-3.1
scooter-black	82.1	97.6	-4.3	82.1	100.0	-4.3
soapbox	79.2	100.0	-2.8	81.0	100.0	-0.4
Average	79.5	95.3	5.8	80.7	95.7	6.1

Table 7: **Result breakdown on DAVIS16 validation sequences.** (*M*), (*R*), and (*D*) are mean, recall and decay of IoU, respectively

Sequence	*w/o CRF* $\mathcal{J}$(M)	*w/ CRF* $\mathcal{J}$(M)
drift	86.1	86.5
birdfall	67.8	57.1
girl	84.5	86.3
cheetah	57.0	50.8
worm	83.7	84.0
parachute	90.6	93.2
monkeydog	22.9	22.6
hummingbird	57.3	57.2
soldier	77.4	77.4
bmx	76.4	77.5
frog	84.1	86.7
penguin	77.7	76.8
monkey	75.0	75.8
bird of paradise	92.3	94.0
Seq. Avg.	73.8	73.3
Frame Avg.	78.3	78.9

Table 8: Sequence breakdown on SegTrackv2 dataset.

Sequence	*w/o CRF* $\mathcal{J}$(M)	*w/ CRF* $\mathcal{J}$(M)
camel01	86.8	91.0
cars1	86.9	86.8
cars10	64.6	64.8
cars4	81.5	82.4
cars5	81.6	82.1
cats01	87.7	89.5
cats03	69.4	63.2
cats06	66.5	67.4
dogs01	76.3	75.6
dogs02	85.3	86.4
farm01	90.8	90.5
giraffes01	82.1	83.9
goats01	79.9	83.7
horses02	80.4	83.6
horses04	59.8	60.5
horses05	72.8	74.5
lion01	75.1	75.0
marple12	81.9	81.6
marple2	84.4	85.9
marple4	81.1	82.4
marple6	95.1	95.1
marple7	76.6	77.6
marple9	95.4	96.3
people03	90.1	91.0
people1	85.3	87.2
people2	88.1	89.7
rabbits02	91.2	91.2
rabbits03	81.5	84.4
rabbits04	43.8	44.1
tennis	73.3	74.2
Seq. Avg.	79.8	80.7
Frame Avg.	77.4	78.4

Table 9: Sequence breakdown on FBMS59 dataset

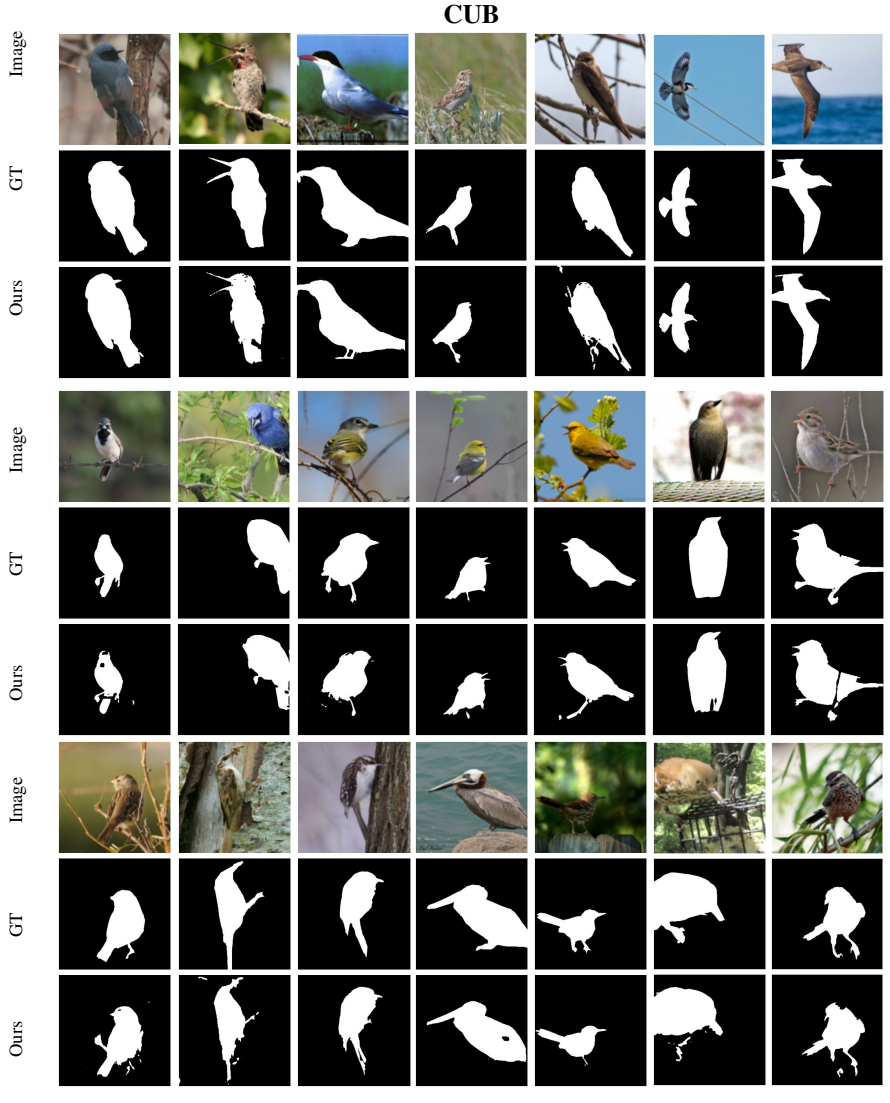

Figure 1: **Qualitative Comparison on CUB**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our method can extract birds in different environments and poses. Our model can segment different species of birds

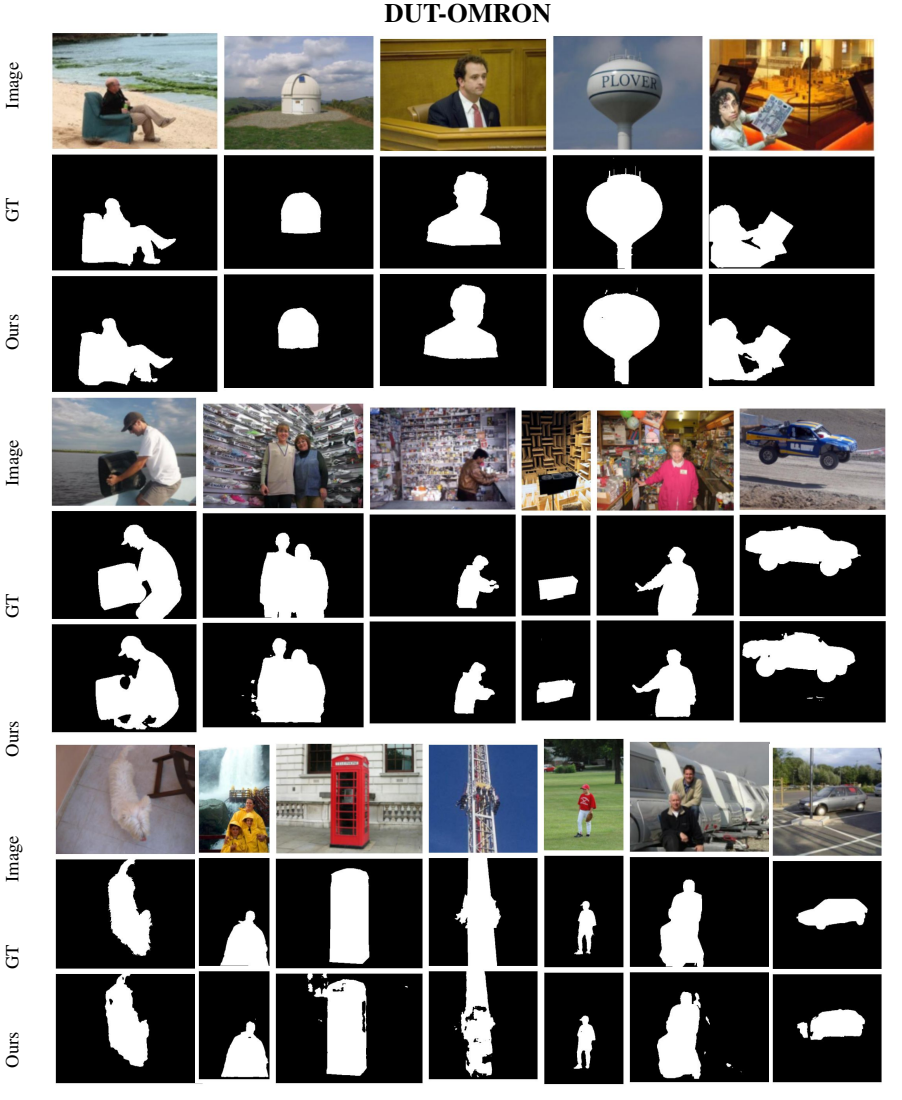

Figure 2: **Qualitative Comparison on DUT-OMRON**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our model can segment both stationary and non-stationary objects and is robust enough to work on a wide range of classes

Figure 3: **Qualitative Comparison on DUTS**. We train our model on a combined dataset of DAVIS, FBMS and STv2. We can segment a wide array of classes. Our model performs well on scenes where multiple objects are in the foreground

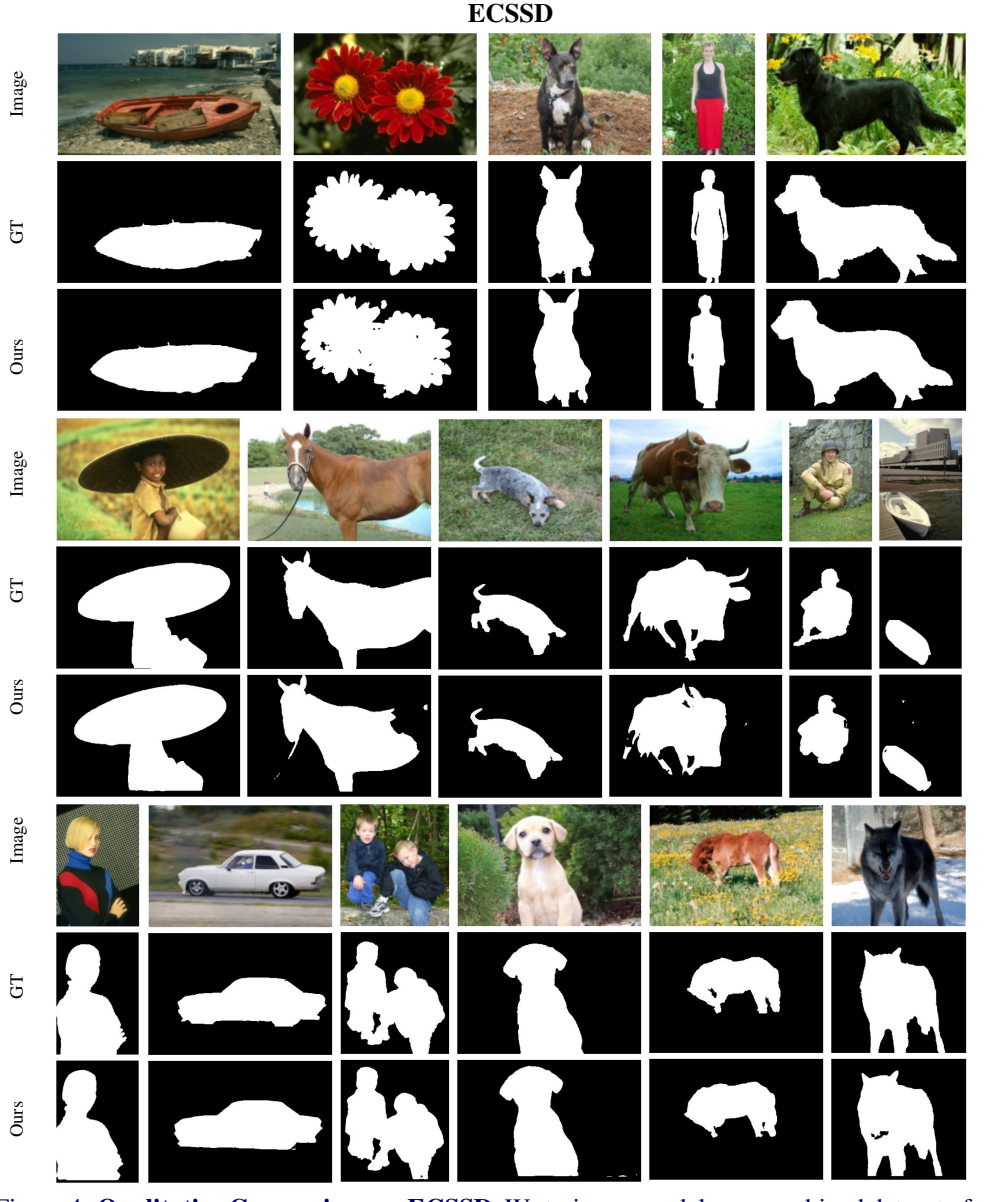

Figure 4: **Qualitative Comparison on ECSSD**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our model can segment objects from different classes in complex poses

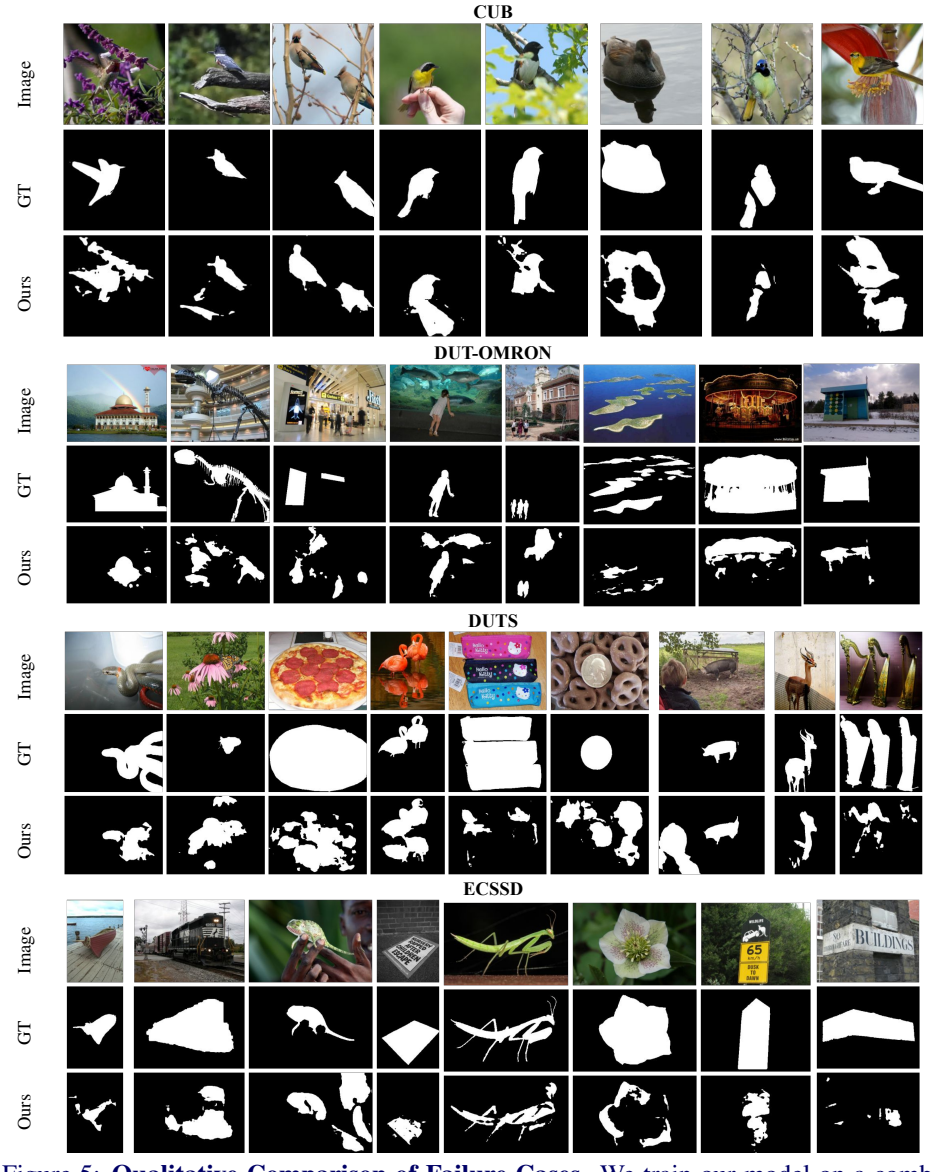

Figure 5: **Qualitative Comparison of Failure Cases**. We train our model on a combined dataset of DAVIS, FBMS and STv2. Our method can extract salient object in various environments. The model has difficulty where the foreground object is ambiguous — when there are multiple prominent objects but only few are annotated as salient object. The model also has issues with predicting the object boundaries well for some instances

References

[1] Yaniv Benny and Lior Wolf. Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. In *European Conference on Computer Vision*, pages 514–530. Springer, 2020.

[2] Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October 2021.

[4] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. *Advances in neural information processing systems*, 32, 2019.

[5] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems*, 2021.

[6] Xingzhe He, Bastian Wandt, and Helge Rhodin. Ganseg: Learning to segment by unsupervised hierarchical image generation. *arXiv preprint arXiv:2112.01036*, 2021.

[7] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.

[8] Asako Kanezaki. Unsupervised image segmentation by backpropagation. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1543–1547. IEEE, 2018.

[9] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] Runtao Liu, Zhirong Wu, Stella Yu, and Stephen Lin. The emergence of objectness: Learning zero-shot segmentation from videos. *Advances in Neural Information Processing Systems*, 34, 2021.

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[12] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8364–8375, June 2022.

[13] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Finding an unsupervised image segmenter in each of your deep generative models. In *International Conference on Learning Representations*, 2022.

[14] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. Em-driven unsupervised learning for efficient motion segmentation. *CoRR*, abs/2201.02074, 2022.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115 (3):211–252, 2015.

[16] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3971–3980, June 2022.

[17] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

[19] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Object segmentation without labels with large-scale generative models. In *International Conference on Machine Learning*, pages 10596–10606. PMLR, 2021.

[20] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14543–14553, June 2022.

[21] Xide Xia and Brian Kulis. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*, 2017.

[22] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177–7188, 2021.

[23] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] Yanchao Yang, Brian Lai, and Stefano Soatto. Dystab: Unsupervised object segmentation via dynamic-static bootstrapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2826–2836, 2021.

[25] Peiyu Yu, Sirui Xie, Xiaojian Ma, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Unsupervised foreground extraction via deep region competition. *Advances in Neural Information Processing Systems*, 34, 2021.