UCOD-DPL: Unsupervised Camouflaged Object Detection via Dynamic Pseudo-label Learning

Supplementary Material

6. Loss Defination

In Sec. 3.2, we employ the binary cross-entropy loss $\mathcal{L}_{\mathrm{BCE}}$ in Eq. (4) and Eq. (9). We assume $\hat{y}_i$ denotes the model's prediction for the i-th labeled data and y_i denotes its label(*i.e.* ground-truth or pseudo-label). The binary cross-entropy loss can be defined as below:

$$\mathcal{L}_{\mathrm{BCE}}(\hat{y}, y) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{15}$$

7. Experiment Details

More Visualization Results. As shown in Fig. 7, we further present additional challenging samples from the CHALEMEON, CAMO, COD10K, and NC4K datasets, corresponding to scenarios involving small objects, large objects, multiple instances, and highly camouflaged samples. From the comparative results, it can be observed that our method demonstrates superior prediction quality in scenarios involving small objects and multiple objects. However, partial segmentation blur still occurs in some samples with complex textures and high background similarity.

Full Quantitative Analysis. As shown in Tab. 5 and Tab. 6, we present the evaluation results of our method across all metrics on the four datasets. We employ the Accuracy(Acc), mean Intersection over Union($mIoU$), S-measure ($\mathcal{S}_m$) [8], mean and weighted F-measure ($\mathcal{F}_\beta^m$, $\mathcal{F}_\beta^\omega$) [31], mean and max E-measure ($\mathcal{E}_\epsilon^m$, $\mathcal{E}_\epsilon^x$) [9], mean absolute error ($\mathcal{M}$) [39] as the evaluation metrics.

Implmentation Details. All experiments are implemented with PyTorch 2.1, Accelerate 1.0.1, and run on a machine with Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz, 512GiB RAM, and 2 NVIDIA Titan A100-40G GPUs. All experiments use the same random seed.

We build the network by using DINOv1 and DINOv2 as the backbone to extract image features, which are then processed by subsequent modules. All backbone parameters remain frozen during the training process. We set the learning rate to $2e-4$ and the batch size to 32, using the AdamW as the optimizer and StepLR(step=25) as the learning rate decay strategy. We train the model for 25 epochs, with the last 5 epochs using only the teacher model's predictions to fine-tune the student model via supervised learning. All inputs are reshaped to 518×518, while the features extracted by the backbone are reshaped to 68×68 during the training and testing process.

Full Ablation Study on Foreground-sizes. We divided the test set by the proportion of test set prospects at 2% intervals, then benchmarked the performance between our method and some previous SOTA methods[44, 63]. The full results are shown at Fig. 8. We alse employ the Accuracy(Acc), mean Intersection over Union($mIoU$), S-measure ($\mathcal{S}_m$) [8], mean and weighted F-measure ($\mathcal{F}_\beta^m$, $\mathcal{F}_\beta^\omega$) [31], mean and max E-measure ($\mathcal{E}_\epsilon^m$, $\mathcal{E}_\epsilon^x$) [9], mean absolute error ($\mathcal{M}$) [39] as the evaluation metrics.

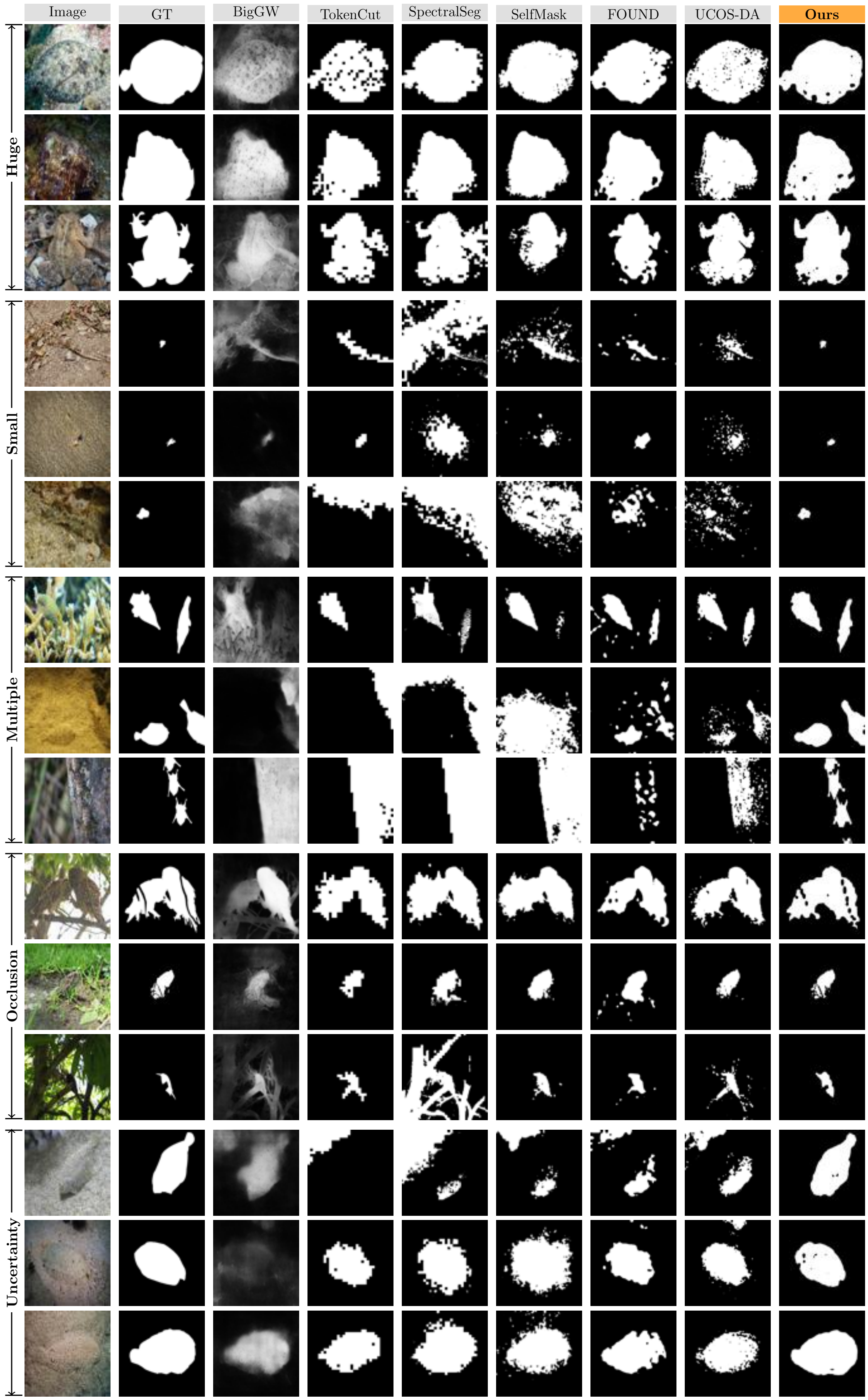

Figure 7. Full visual comparison of our method with other existing methods in challenging scenarios.

Methods	CHAMELEON (87)								CAMO-Test (250)							
	$Acc\uparrow$	$mIOU\uparrow$	$\mathcal{S}_m\uparrow$	$\mathcal{F}_\beta^\omega\uparrow$	$\mathcal{F}_\beta^m\uparrow$	$\mathcal{E}_\phi^m\uparrow$	$\mathcal{E}_\phi^x\uparrow$	$\mathcal{M}\downarrow$	$Acc\uparrow$	$mIOU\uparrow$	$\mathcal{S}_m\uparrow$	$\mathcal{F}_\beta^\omega\uparrow$	$\mathcal{F}_\beta^m\uparrow$	$\mathcal{E}_\phi^m\uparrow$	$\mathcal{E}_\phi^x\uparrow$	$\mathcal{M}\downarrow$
Fully-Supervised Methods																
SINet_{20}[10]	-	-	.872	.806	.827	.946	-	.034	-	-	.751	.606	.675	.771	-	.100
$\text{C}^2\text{FNet}_{21}$[47]	-	-	.888	.828	.844	.946	-	.032	-	-	.796	.719	.762	.864	-	.080
MGL-R_{21}[61]	-	-	.893	.812	.834	.941	-	.030	-	-	.775	.673	.726	.842	-	.088
UGTR_{21}[58]	-	-	.887	.794	.819	.940	-	.031	-	-	.784	.684	.735	.851	-	.086
BGNet_{22}[48]	-	-	.901	.851	.860	.954	-	.027	-	-	.812	.749	.789	.870	-	.073
ZoomNet_{22}[37]	-	-	.902	.845	.864	.958	-	.023	-	-	.820	.752	.794	.878	-	.066
SINetv2_{22}[12]	-	-	.888	.816	.835	.961	-	.030	-	-	.820	.743	.782	.882	-	.070
HitNet_{23}[22]	-	-	<u>.921</u>	<u>.897</u>	<u>.900</u>	**.972**	-	<u>.019</u>	-	-	.849	<u>.809</u>	<u>.831</u>	<u>.906</u>	-	.055
FSPNet_{23}[23]	-	-	.908	.851	.867	.965	-	.023	-	-	<u>.856</u>	.799	.830	.899	<u>.928</u>	<u>.050</u>
BiRefNet_{24}[64]	-	-	**.929**	**.911**	**.922**	<u>.968</u>	-	**.016**	-	-	**.932**	**.914**	**.922**	**.974**	**.959**	**.015**
Semi-Supervised Methods																
CamoTeacher_{24}(1%)[27]	-	-	.652	.472	.558	.714	.762	.093	-	-	.621	.456	.545	.669	.736	.136
CamoTeacher_{24}(5%)[27]	-	-	.729	.587	.656	.785	.822	.070	-	-	.669	.523	.601	.711	.775	.122
CamoTeacher_{24}(10%)[27]	-	-	.756	.617	.684	.813	.851	.065	-	-	.701	.560	.742	.795	.795	.112
SCOD-ND_{24}(10%)[13]	-	-	.850	.773	-	.928	-	.036	-	-	.789	.732	-	.859	-	.077
Unupervised Methods																
BigGW_{21} [51]	.807	.267	.547	.244	.294	.527	.662	.257	.775	.322	.565	.299	.349	.528	.678	.282
TokenCut_{22}[53]	.868	.436	.654	.496	.536	.740	.743	.132	.837	.431	.633	.498	.543	.706	.708	.163
TokenCut_{22} w/B.S.[53]	.871	.415	.655	.351	.393	.582	.734	.169	.838	.422	.639	.383	.434	.595	.699	.195
SpectralSeg_{22}[32]	.780	.381	.575	.410	.440	.628	.638	.220	.765	.411	.579	.450	.481	.648	.658	.235
SelfMask_{22}[43]	.825	.396	.619	.436	.481	.675	.726	.176	.813	.418	.617	.483	.536	.698	.713	.176
SelfMask_{22} w/U.B.[43]	.832	.406	.629	.447	.491	.683	.734	.169	.819	.430	.627	.495	.547	.708	.724	.182
$\text{FOUND}_{23-DINOv1}$[44]	.905	.468	.684	.542	.590	.810	.812	.095	.871	.505	.685	.584	.633	.782	.784	.129
*$\text{FOUND}_{23\text{-}DINOv2}$[44]	<u>.943</u>	<u>.691</u>	<u>.829</u>	<u>.757</u>	<u>.781</u>	<u>.911</u>	<u>.921</u>	<u>.040</u>	<u>.905</u>	<u>.628</u>	<u>.770</u>	<u>.704</u>	<u>.740</u>	<u>.849</u>	<u>.851</u>	<u>.090</u>
$\text{UCOS-DA}_{23-DINOv1}$[63]	.905	.525	.715	.591	.629	.802	.804	.095	.873	.528	.701	.606	.646	.784	.768	.127
*$\text{UCOS-DA}_{23-DINOv2}$[63]	.840	.455	.750	.639	.666	.808	.692	.091	.812	.470	.702	.604	.633	.751	.692	.148
Ours$_{DINOv1}$	.928	.529	.734	.625	.680	.854	.856	.072	.892	.508	.706	.621	.689	.801	.803	.108
Ours$_{DINOv2}$	**.969**	**.753**	**.864**	**.825**	**.838**	**.931**	**.933**	**.031**	**.923**	**.652**	**.793**	**.747**	**.779**	**.862**	**.865**	**.077**

Table 5. **Full comparison of our methods with recent methods on CHAMELEON and CAMO test datasets.** We compared our proposed methods with competing unsupervised, semi-supervised, and full-supervised methods. **Bold** indicates the best result in group settings, and <u>underline</u> indicates the second-best result. * denotes the version that reimplemented by us.

Methods	COD10K-Test (2,026)								NC4K (4,121)							
	$Acc\uparrow$	$mIOU\uparrow$	$\mathcal{S}_m\uparrow$	$\mathcal{F}_\beta^\omega\uparrow$	$\mathcal{F}_\beta^m\uparrow$	$\mathcal{E}_\phi^m\uparrow$	$\mathcal{E}_\phi^x\uparrow$	$\mathcal{M}\downarrow$	$Acc\uparrow$	$mIOU\uparrow$	$\mathcal{S}_m\uparrow$	$\mathcal{F}_\beta^\omega\uparrow$	$\mathcal{F}_\beta^m\uparrow$	$\mathcal{E}_\phi^m\uparrow$	$\mathcal{E}_\phi^x\uparrow$	$\mathcal{M}\downarrow$
Fully-Supervised Methods																
SINet_{20}[10]	-	-	.771	.551	.634	.806	-	.051	-	-	.808	.723	.769	.871	-	.058
$\text{C}^2\text{FNet}_{21}$[47]	-	-	.813	.686	.723	.900	-	.036	-	-	.838	.762	.794	.904	-	.049
MGL-R_{21}[61]	-	-	.814	.666	.710	.890	-	.035	-	-	.833	.739	.782	.893	-	.053
UGTR_{21}[58]	-	-	.817	.666	.711	.890	-	.036	-	-	.839	.746	.787	.899	-	.052
BGNet_{22}[48]	-	-	.831	.722	.753	.901	-	.033	-	-	.851	.788	.820	.907	-	.044
ZoomNet_{22}[37]	-	-	.838	.729	.766	.888	-	.029	-	-	.853	.784	.818	.896	-	.043
SINetv2_{22}[12]	-	-	.815	.680	.718	.887	-	.037	-	-	.847	.770	.805	.903	-	.048
HitNet_{23}[22]	-	-	.871	.806	.823	.935	-	.023	-	-	.875	.834	.853	.926	-	.037
FSPNet_{23}[23]	-	-	.851	.735	.769	.895	.930	.026	-	-	.879	.816	.843	.915	.937	.035
BiRefNet_{24}[64]	-	-	**.913**	**.874**	**.888**	**.960**	**.967**	**.014**	-	-	**.914**	**.894**	**.909**	**.953**	**.960**	**.023**
Semi-Supervised Methods																
CamoTeacher_{24}(1%)[27]	-	-	.699	.517	.582	.788	.797	.062	-	-	.718	.599	.675	.779	.814	.090
CamoTeacher_{24}(5%)[27]	-	-	.745	.583	.644	.827	.840	.050	-	-	.777	.677	.739	.834	.859	.071
CamoTeacher_{24}(10%)[27]	-	-	.759	.594	.652	.836	.854	.049	-	-	.791	.687	.746	.842	.868	.068
SCOD-ND_{24}(10%)[13]	-	-	.819	.725	-	.891	-	.033	-	-	.838	.787	-	.903	-	.046
Unupervised Methods																
BigGW_{21} [51]	798	.236	.528	.185	.246	.497	.670	.261	.814	.382	.608	.319	.391	.565	.714	.246
TokenCut_{22}[53]	.897	.415	.658	.469	.502	.735	.740	.103	.899	.546	.725	.615	.649	.802	.806	.101
TokenCut_{22} w/B.S.[53]	.903	.423	.666	.334	.399	.609	.739	.127	.904	.561	.735	.478	.547	.683	.807	.133
SpectralSeg_{22}[32]	.807	.331	.575	.360	.388	.595	.606	.193	.841	.495	.669	.535	.562	.719	.729	.159
SelfMask_{22}[43]	.870	.388	.637	.431	.469	.679	.718	.131	.887	.529	.716	.593	.634	.777	.796	.114
SelfMask_{22} w/U.B.[43]	.875	.397	.645	.440	.478	.687	.728	.125	.891	.538	.723	.601	.642	.784	.803	.110
$\text{FOUND}_{23-DINOv1}$[44]	.915	.428	.670	.482	.520	.751	.753	.085	.916	.566	.741	.637	.674	.824	.827	.084
*$\text{FOUND}_{23\text{-}DINOv2}$[44]	.946	.574	.767	.641	.668	.847	.850	.045	.939	.679	.816	.756	.783	.893	.896	.052
$\text{UCOS-DA}_{23-DINOv1}$[63]	.914	.462	.689	.513	.546	.740	.741	.086	.915	.590	.755	.656	.689	.819	.822	.085
*$\text{UCOS-DA}_{23-DINOv2}$[63]	.882	.430	.655	.467	.495	.687	.689	.120	.897	.570	.731	.617	.644	.785	.787	.103
Ours$_{DINOv1}$	.941	.492	.727	.577	.627	.822	.824	.059	.926	.573	.761	.680	.737	.851	.853	.074
Ours$_{DINOv2}$	**.969**	**.680**	**.834**	**.763**	**.779**	**.916**	**.918**	**.031**	**.958**	**.734**	**.850**	**.818**	**.835**	**.923**	**.925**	**.043**

Table 6. **Full comparison of our methods with recent methods on COD10K and NC4K test datasets.** We compared our proposed methods with competing unsupervised, semi-supervised, and full-supervised methods. **Bold** indicates the best result in group settings, and underline indicates the second-best result. * denotes the version that reimplemented by us.

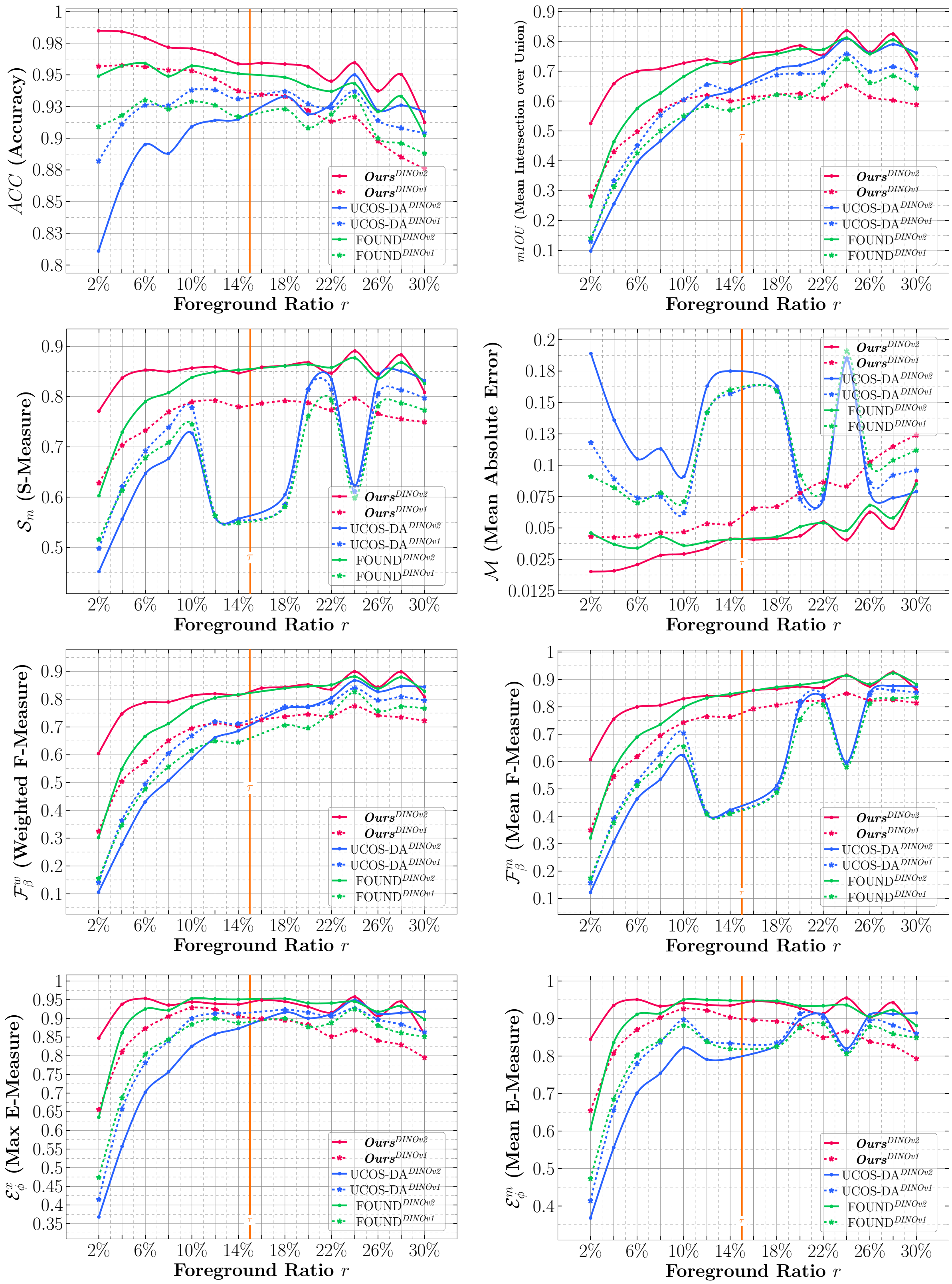

Figure 8. **Performance comparison for different foreground sizes on COD10K-Test dataset.**