3D-Aware Multi-Task Learning with Cross-View Correlations for Dense Scene Understanding

Supplementary Material

A6. More Details

A6.1. Implementation Details

We apply our method to different state-of-the-art MTL methods, including SAK [43], RADIO [49] from Lu *et al*. [43] and DINOv3 [56], by attaching the cross-view module (CvM) to their encoder. For all experiments, we follow the identical training and evaluation protocol of prior work [43]. We implement our model in PyTorch [47] and use AdamW [42] as optimizer with a learning rate of 2×10^{-5} and a weight decay rate of 1×10^{-6}. Polynomial learning rate scheduler is used to dynamically adjust the learning rate. We use a batch size of 4 and train each model for 40000 steps. The image size is 448×576 for NYUv2 and 512×512 for PASCAL-Context. We use the same loss functions and loss weights as in Lu *et al*. [43, 61, 72]: cross-entropy loss for segmentation, human parsing, saliency and boundary detection, L1 loss for depth and normal estimation. Across all experiments, we use ViT-L as backbone for MTL encoder, and utilize multi-scale features for vision transformer, i.e., intermediate features from layer 5, 12, 18, 24. More details on the design of CvM are presented in Sec. A6.2.

A6.2. Details for CvM

In our CvM, we implement a ResNet-style [22] convolutional network as the spatial-aware encoder to extract geometry-sensitive features from multi-view RGB inputs. The encoder consists of three residual blocks, progressively reducing spatial resolution while increasing channel dimensions, ultimately producing a 128-dimensional feature map at $1/8$ the input resolution. This spatial feature is then fed into the multi-view transformer module.

The multi-view transformer comprises six layers of self-attention and cross-view attention, each employing the Swin Transformer's [40] window-based attention mechanism to preserve local context while enabling efficient cross-view communication. To better align the cross-view enhanced features with the MTL feature space, we remove the output normalization of the final cross-attention layer and instead append a lightweight SwiGLU-based [54] adapter module, which consists of a gated MLP layer. The resulting cross-view enhanced features are subsequently aligned using the camera intrinsics and relative poses between views to construct a cost volume. Specifically, given a set of depth candidates $d_1, \ldots, d_L$ sampled in inverse-depth space over a range of 0.0001 to 10 meters, we follow a differentiable feature warping strategy to reproject the features from neighboring views onto the coordinate frame of the reference view. Concretely, for each pixel location in the neighboring view, we back-project it into a 3D point at each hypothesized depth using its camera intrinsics. These 3D points are then transformed into the coordinate system of the reference view using the relative camera pose. The transformed 3D points are subsequently reprojected into the reference image plane using its intrinsics, yielding a dense sampling grid across depth planes. The resulting warped features are used to construct a volumetric cost volume, which encodes the view-wise matching information across different depth planes and serves as a strong 3D-aware cue for subsequent MTL prediction.

Then, both the cross-view enhanced features and cost volume are upsampled by a learned $4\times$ upsampling module. These upsampled features are concatenated with the multi-scale features from the MTL encoder and fused within the task-specific decoder heads. Finally, a linear layer takes the fused 3D-aware multi-task feature as input and regresses the MTL predictions for each task.

For single-view setting, we follow UniMatch [67] and set the relative camera transformation, including both the intrinsic and extrinsic matrices, to the identity. Consequently, the induced warping grid becomes the identity mapping.

A7. Additional Results

A7.1. Detailed MTL Performance for Ablation Study

The detailed task-specific results for our ablation study on *extracting spatial-aware features* (Tab. 5 in the main paper), *number of Depth Candidates* L (Tab. 6 in the main paper) and *number of views* (Tab. 7 in the main paper) are presented in the following Tab. A8, Tab. A9 and Tab. A10, respectively.

Spatial-aware Encoder	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
MTL Encoder	**65.34**	0.3847	15.52	80.54	18.84
MTL Encoder + LoRA [24]	64.83	**0.3760**	15.40	80.89	18.87
MTL Encoder + Adapter [43]	**65.34**	0.3814	**15.30**	80.85	18.98
Ours	65.27	0.3836	15.35	**81.69**	**19.05**

Table A8. Detailed MTL performance for comparisons of various spatial-aware feature extraction methods in cross-view module on NYUv2. ΔMTL is computed using single-task learning "STL" in Tab. 2 in the main paper as baseline.

Detailed Results for Ablation on Extracting Spatial-aware Features. As shown in Tab. A8, although different designs for extracting spatial-aware features exhibit varying performance across tasks, our CNN-based design achieves the highest overall MTL performance. This architecture-agnostic and non-intrusive design also makes CvM readily applicable to a wider range of MTL encoders, highlighting its practical flexibility and generalizability.

#Depth Candidates	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
96	64.84	0.3827	15.32	81.54	18.88
128	65.27	0.3836	15.35	**81.69**	19.05
256	64.21	0.3832	**15.29**	81.58	18.62
384	64.08	0.3835	15.34	81.51	18.46
512	**65.28**	**0.3778**	15.37	81.57	**19.25**

Table A9. Detailed MTL performance for comparisons of various number of depth candidates on NYUv2. ΔMTL is computed using single-task learning "STL" in Tab. 2 in the main paper as baseline.

Detailed Results for Ablation on Number of Depth Candidates. In Tab. A9, increasing the number of depth candidates to 512 significantly improves depth estimation performance, reducing the RMSE from 0.3836 (with 128 candidates) to 0.3778, while maintaining comparable performance on the other tasks. However, such a fine-grained discretization of the depth range introduces substantial computational overhead due to the increased cost of feature warping during cost volume construction. Considering the trade-off between performance and efficiency, setting $L = 128$ offers an optimal balance, which aligns with commonly adopted settings in recent 3D reconstruction pipelines such as MVSplat [12] and DepthSplat [68].

Detailed Results for Ablation on Number of Views. In Tab. A10, we observe that increasing the number of input views from 2 to 3 leads to improved overall MTL performance. However, further increasing the number of views to 4 results in a slight performance drop. We attribute this to the increased complexity of learning cross-view correlations from multiple views, which, when combined with the inherent challenge of balancing multiple tasks in MTL, may hinder effective optimization. Moreover, as NYUv2 video sequences lack ground-truth camera poses, we rely on poses estimated via COLMAP [52], which may introduce noise. Using more views could amplify such pose estimation errors, negatively impacting the quality of learned geometric features. Future work may explore pose-free alternatives or incorporate view-relative pose prediction to enable more robust multi-view training.

A7.2. Results with ViT-B Backbones

In this section, we provide results with ViT-B backbones to evaluate the generality of our method across different backbone capacities and to complement the main results reported with ViT-L.

Input Views	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
2	**65.27**	0.3836	15.35	81.69	19.05
3	65.17	**0.3827**	**15.24**	**81.72**	**19.20**
4	64.97	0.3913	15.28	81.61	18.63

Table A10. Detailed MTL performance for results of various number of views. ΔMTL is computed using single-task learning "STL" in Tab. 2 in the main paper as baseline.

MTL with Multiple Views. We first conduct multi-view MTL experiments following the setup of Tab. 1 in the main paper. Results are presented in Tab. A11. After reducing the number of parameters in the MTL encoder, it becomes increasingly difficult for the model to directly learn effective cross-view correlations and 3D awareness by simply introducing multi-view data during training. Although this strategy still improves overall MTL performance, it proves inefficient for injecting 3D priors and shows limited benefits across individual tasks.

In contrast, our CvM makes more effective use of the multi-view data, enabling efficient injection of 3D awareness into the MTL model. Specifically, our CvM leads to an MTL performance gain of +2.31 and +3.19 for SAK and DINOv3, respectively, compared to their baselines trained without video data. Notably, when applied to DINOv3, our method not only surpasses its multi-view trained counterpart but also outperforms 3DMTL [32] across all tasks. These results further confirm that CvM learns cross-view correlations more effectively and consistently enhances the performance of MTL models by introducing 3D geometric awareness.

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
SAK [43] *w/o video*	**59.93**	0.4942	17.60	78.60	0.00
SAK [43]	59.41	0.4718	17.65	78.38	0.78
Ours	58.97	**0.4534**	**17.40**	**79.74**	**2.31**
DINOv3 [56] *w/o video*	59.73	0.4650	16.80	78.53	0.00
DINOv3 [56]	59.72	0.4450	16.90	78.40	0.88
3DMTL*	59.65	0.4403	16.72	78.72	1.47
Ours	**60.74**	**0.4263**	**16.66**	**80.03**	**3.19**

Table A11. Quantitative comparison of our method with ViT-B backbone on NYUv2 dataset + extra video frames with multiple views. *: Code for 3DMTL [32] is not available and we reproduce 3DMTL [32] with DINOv3 backbone. ΔMTL is computed using "SAK [43] *w/o video*" and "DINOv3 [56] *w/o video*" as baseline, respectively.

Comparison with SotAs. When integrating our CvM into state-of-the-art MTL frameworks with ViT-B backbones, following the same setting of Tab. 2 and Tab. 3 in the main paper, we observe a similar trend of performance improvement as with ViT-L. On the NYUv2 dataset, our method consistently enhances the performance of both RADIO [49] and DINOv3 [56] across all tasks. It also sur-

passes SAK [43] on three out of four tasks, while achieving comparable segmentation performance. On the PASCAL-Context dataset, our CvM again delivers comprehensive gains for all three MTL encoders, demonstrating a similar trend to the ViT-L backbone results. Detailed comparisons are provided in Tab. A12 and Tab. A13.

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
STL	51.15	0.5792	19.77	77.35	0.00
MTL	49.27	0.5823	19.92	75.88	-1.72
BFCI [78]	51.14	0.5186	18.92	77.98	3.89
TSP [64]	51.22	0.5301	18.78	76.90	3.26
InvPT [72]	50.30	0.5367	19.00	77.60	2.47
RADIO [49]	55.03	0.5186	18.49	77.97	6.33
Ours	**55.96**	**0.4970**	**18.36**	**79.35**	**8.32**
SAK [43]	**59.93**	0.4942	17.60	78.60	11.11
Ours	59.60	**0.4535**	**17.34**	**79.95**	**13.47**
DINOv3 [56]	59.73	0.4650	16.80	78.53	13.26
Ours	**60.61**	**0.4376**	**16.63**	**80.38**	**15.69**

Table A12. Quantitative comparison of our method with ViT-B backbone to the SotA methods on NYUv2 dataset. ΔMTL is computed using single-task learning "STL" as baseline.

Method	Seg. (mIoU) ↑	PartSeg (mIoU) ↑	Sal (maxF) ↑	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
STL	80.25	70.54	84.54	13.57	74.22	0.00
MTL	76.76	65.26	84.39	13.98	70.37	-4.04
TaskExpert [74]	78.45	67.38	84.96	13.55	72.30	-1.73
BFCI [78]	77.98	68.19	85.06	13.48	72.98	-1.31
MLoRE [70]	79.26	67.82	85.31	13.65	74.69	-0.83
InvPT [72]	77.33	66.62	85.14	13.78	73.20	-2.28
RADIO [49]	78.06	68.13	85.18	13.59	72.64	-1.53
Ours	**78.21**	**69.20**	**85.20**	**13.50**	**75.82**	**-0.20**
SAK [43]	81.88	74.30	84.79	14.02	74.09	0.83
Ours	**81.94**	**75.22**	**84.90**	**13.72**	**77.76**	**2.57**
DINOv3 [56]	81.46	74.11	84.71	13.81	73.94	2.52
Ours	**82.10**	**75.06**	**85.18**	**13.67**	**77.64**	**2.69**

Table A13. Quantitative comparison of our method with ViT-B backbone to the SotA methods on PASCAL-Context dataset. ΔMTL is computed using single-task learning "STL" as baseline.

A7.3. Comparison with 3D Foundation Models

While 3D foundation models (FMs) target geometry from massive data to solve scene reconstruction tasks, we focus on general MTL (geometry & semantics) where data are scarce and costly to obtain. Thus, FMs and our CvM are orthogonal and complementary.

However, it would be interesting to evaluate how geometric foundation models perform on general MTL and how our CvM can improve their performance. Therefore, we added comparisons with FMs, VGGT [62] and DepthAnything3 (DA3) [34], by using their features as MTL encoder and adding heads and our CvM. Specifically, for DA3, we adopt the DA3Metric-Large model, as its model scale and task setup are more comparable to the MTL setting. As shown in Tab. A14, directly using foundation models as MTL encoders yields worse performance than Ours, while incorporating CvM consistently improves their MTL results. This verifies that CvM is complementary to 3D FMs for MTL.

Method	Seg.(mIoU) ↑	Depth(RMSE) ↓	Normal(mErr) ↓	Boundary(odsF) ↑	ΔMTL ↑
DINOv3 [56] *w/o video*	63.68	0.4113	15.53	80.10	0.00
DINOv3 [56]	64.03	0.3954	15.35	80.52	1.52
Ours	**65.27**	**0.3836**	15.35	**81.69**	**3.09**
VGGT [62]	64.82	0.4002	15.78	80.94	0.98
VGGT + CvM (Ours)	**65.16**	**0.3592**	15.49	**81.78**	**4.34**
DA3 [34]	61.38	0.4112	16.30	80.65	-1.96
DA3 + CvM (Ours)	**61.87**	**0.4004**	16.15	**81.46**	**-0.62**

Table A14. Quantitative comparison of our method with 3D FMs on NYUv2 dataset + extra video frames with multiple views. ΔMTL is computed using "DINOv3 [56] *w/o video*" as baseline.

A7.4. Cross-task Consistency Metrics

In this paper, we follow exactly the same evaluation metrics as in prior MTL works [43, 72]. We provide qualitative predictions which verify that ours obtains better cross-task consistency, e.g. curtain between Seg. and Edge highlighted by red arrows in Fig. 1.

To quantify cross-task consistency, we further report the angular error between depth-derived normals and predicted normals on NYUv2. Our method achieves a lower error than DINOv3: $28.00°$ vs. $30.42°$. This indicates that CvM improves cross-task consistency beyond task-wise accuracy.

A7.5. More Ablations on CvM

To better understand the contribution of the cross-view enhanced features $\boldsymbol{F}_i$, we include an additional ablation study for "Ours *w/o CV*" in Tab. A15, which removes the cost volume and uses only the cross-view enhanced features. As we can see, using cross-view enhanced features or cost volume alone improves performance, while the full CvM achieves the best results.

To verify that the performance gains brought by CvM arise from its effective design rather than merely from increased model capacity, we also introduce a new baseline that removes CvM and instead adds Swin-style self-attention layers (∼5M) after the monocular backbone. As shown in Tab. A15, this baseline underperforms CvM on all tasks, indicating that the improvement is not due to the additional parameters alone, but to the proposed cross-view correlation design.

A8. Computational Cost Analysis

We analyze the computational overhead introduced by our CvM by measuring the forward-pass FLOPs on the NYUv2 dataset with an input resolution of 448×576. Specifically,

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
Ours *w/o CV & CF*	64.03	0.3954	15.35	80.52	17.57
Ours *w/o CF*	64.86	0.3853	15.33	81.18	18.65
Ours *w/o CV*	64.69	0.3856	**15.32**	81.57	18.69
DINOv3 + Self-Attn	64.68	0.3902	15.55	80.07	18.15
Ours	**65.27**	**0.3836**	15.35	**81.69**	**19.05**

Table A15. Ablation study on cross-view module. "Ours *w/o CV & CF*" is the baseline without our cross-view module. "Ours *w/o CF*" and "Ours *w/o CV*" indicate our method that only uses cost volume or cross-view enhanced feature. "DINOv3 + Self-Attn" replaces the whole cross-view module with self-attention layers of comparable size. "Ours" is our full model that uses both cost volume and cross-view enhanced feature. ΔMTL is computed using "STL" in Tab. 2 as baseline.

Module	Params. (M)	FLOPs (T)	Inference Latency (ms)	Peak Memory
Cost Volume	-	0.02	2.22 ± 0.06	119.09 MB
CvM	∼5	0.27	10.15 ± 0.06	382.18 MB
$\text{SAK}_{\text{Encoder}}$	∼350	1.42	39.73 ± 0.38	1707.70 MB
$\text{DINOv3}_{\text{Encoder}}$	300	1.23	40.15 ± 0.15	1347.06 MB

Table A16. Computational cost analysis for CvM with ViT-L backbone. This Table contains number of parameters for MTL encoder and CvM, and FLOPs, Inference Latency and Peak Memory for a single forward on NYUv2 dataset.

we evaluate the FLOPs, inference latency, and peak memory for the SAK-based [43] and DINOv3-based [56] MTL encoders with ViT-L backbone and for our CvM, under the same experimental setting used in the main paper for ViT-L backbone with two input views and a batch size of 1. The FLOPs for the multi-teacher distillation module in SAK is excluded in the calculation. The results are summarized in Tab. A16.

When integrated into MTL encoders, our CvM operates at $\frac{1}{8}$ of the input resolution, thereby introducing only minimal computational overhead. Specifically, it adds 0.27 TFLOPs, corresponding to an approximately 20% increase over the encoder's original computational cost, which remains modest in practice. Despite the increased compute, our CvM yields significant performance gains across all tasks, as demonstrated in both quantitative and qualitative evaluations. This trade-off reflects a favorable balance between efficiency and accuracy: the added cost primarily stems from the multi-view transformer and cost volume construction, which inject valuable geometric priors and 3D consistency into the MTL predictions. Furthermore, our CvM is designed as a modular, lightweight extension that can be appended to any existing MTL encoder without requiring architectural changes. Compared to prior work such as 3DMTL [32], which incurs a similar level of computational overhead, our CvM provides a more practical solution with better performance for integrating 3D awareness into dense prediction pipelines.

A9. More Visualizations

We provide additional qualitative results on the NYUv2 and PASCAL-Context datasets. Fig. A4 presents a visualization sample from NYUv2, while Fig. A5 and Fig. A6 show two examples from the PASCAL-Context dataset. Since PASCAL-Context does not include multi-view video data, we adopt the single-view training setting for these experiments.

Despite the absence of video data for supervising the CvM module, our method still demonstrates clear advantages in semantic tasks, and consistently produces higher-quality predictions for geometric tasks. As shown in Fig. A5, for semantic segmentation and human part segmentation, SAK [43] and DINOv3 [56] struggle to distinguish the background from fine-grained regions such as the subject's arms and legs, while our model successfully recovers these areas. In the saliency task, the traditional MTL model almost collapses the arm into a thin strip, whereas our method preserves the structural integrity of the limb. For edge and surface normal predictions, our CvM also achieves more accurate results, producing high-quality outputs with sharper boundaries and reduced ambiguity around the human body and the control bar of the glider. These results further validate the effectiveness of our CvM and highlight its generalization in single-view settings.

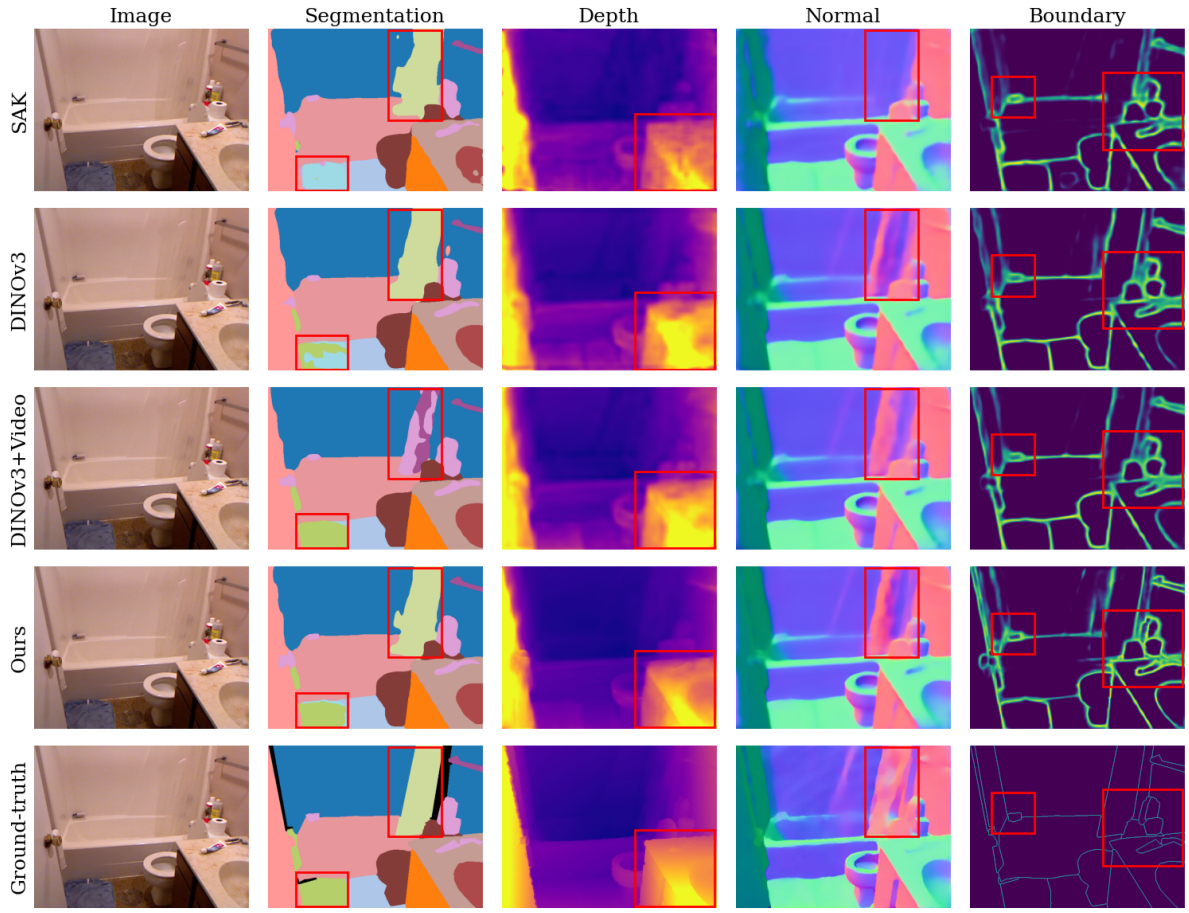

Figure A4. **Qualitative Comparisons on NYUv2.** The first column shows the RGB image, while the remaining columns present either the ground truth or model predictions. The last row shows the ground-truth of four tasks. The first to the fourth rows show the predictions of SAK, DINOv3, DINOv3 trained with videos as multi-view data, and our method, respectively.

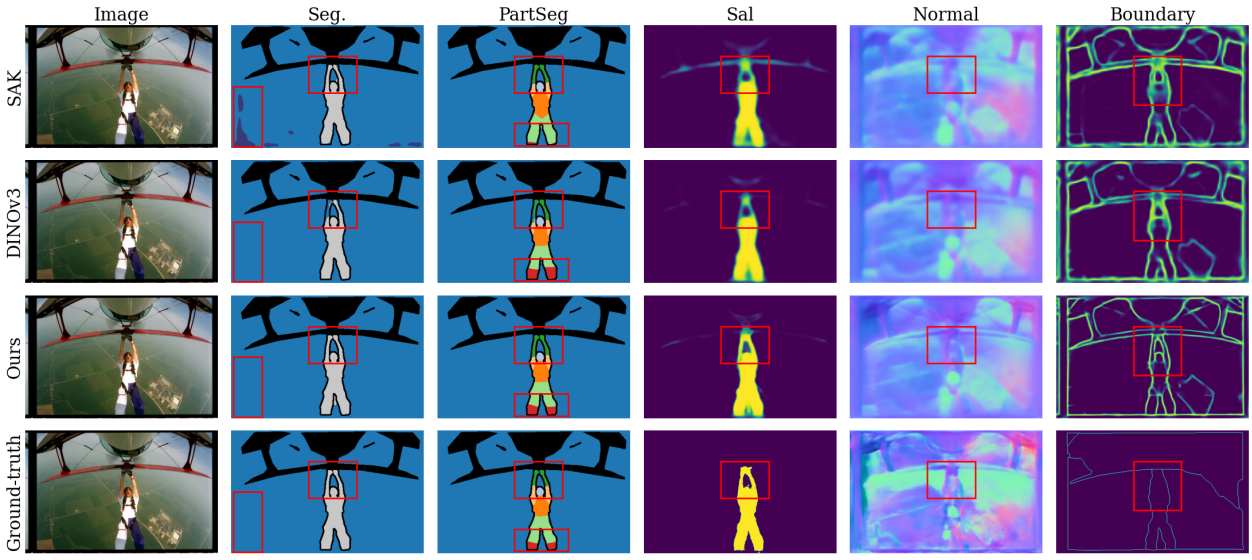

Figure A5. **Qualitative Comparisons on PASCAL-Context.** The first column shows the RGB image, while the remaining columns present either the ground truth or model predictions. The last row shows the ground-truth of five tasks. The first to the third rows show the predictions of SAK, DINOv3, and our method, respectively.

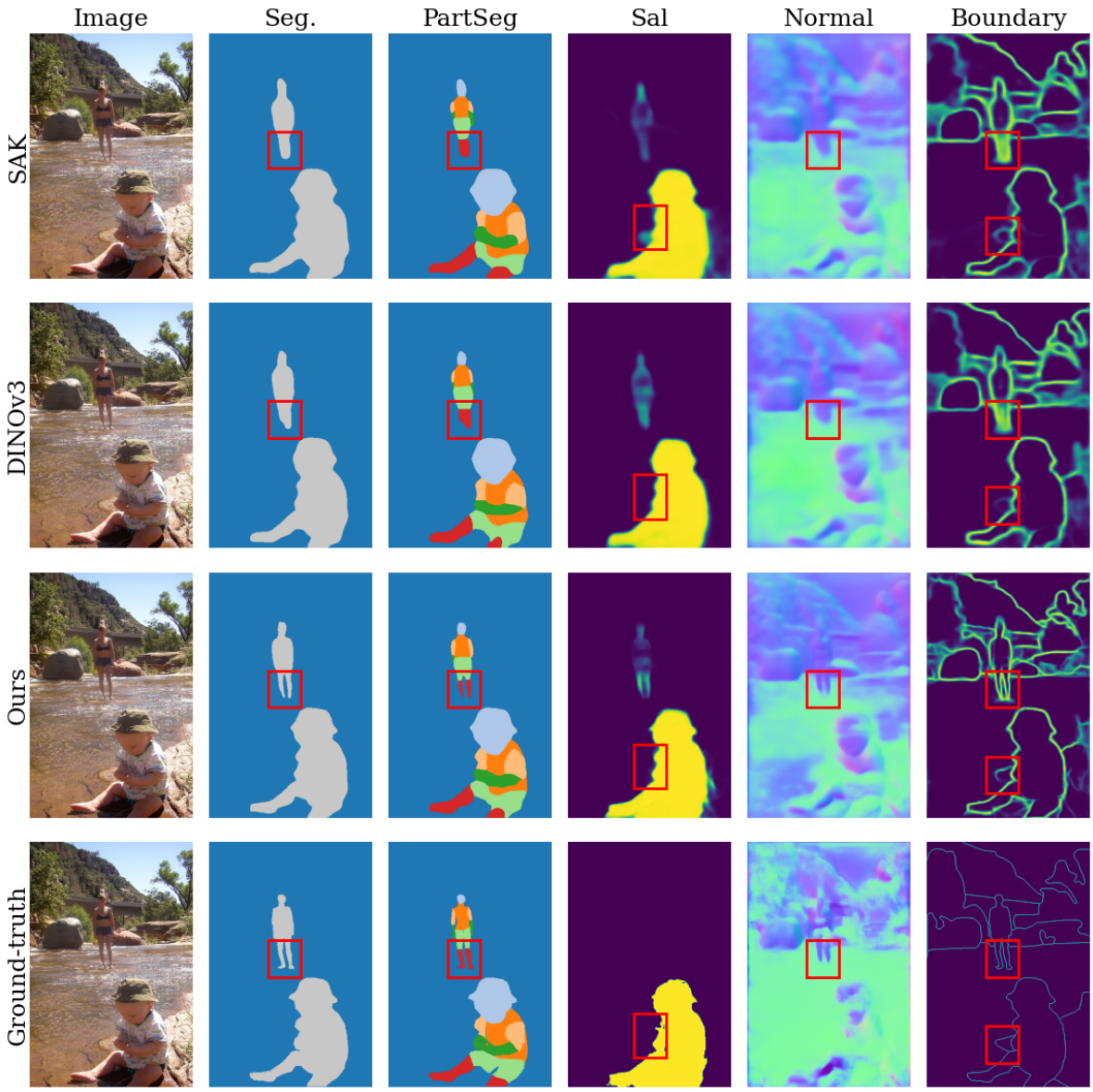

Figure A6. **Qualitative Comparisons on PASCAL-Context.** The first column shows the RGB image, while the remaining columns present either the ground truth or model predictions. The last row shows the ground-truth of five tasks. The first to the third rows show the predictions of SAK, DINOv3, and our method, respectively.