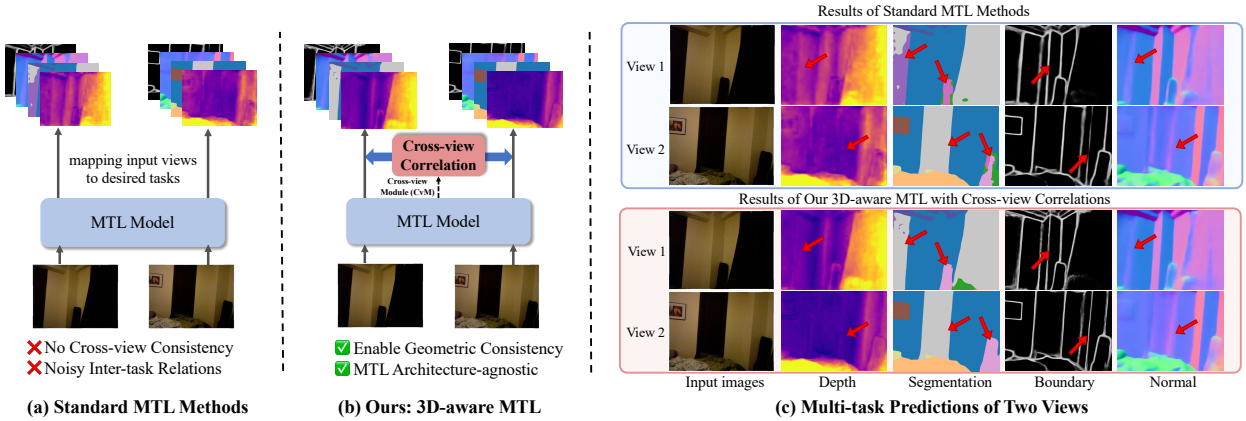
3D-Aware Multi-Task Learning with Cross-View Correlations for Dense Scene Understanding

Anonymous CVPR submission

Paper ID xxxxx



(a) Standard MTL Methods **(b) Ours: 3D-aware MTL** **(c) Multi-task Predictions of Two Views**

Figure 1. **Comparison between standard MTL and our 3D-aware MTL framework.** (a) Standard MTL relies solely on 2D per-pixel supervision, while (b) our approach incorporates geometric consistency through a lightweight cross-view module (CvM). (c) Using DINOv3 [56] as the encoder, standard MTL (top) shows cross-view ambiguities (highlighted by arrows), e.g., inconsistent curtain segmentation, leading to reduced inter-task coherence. In contrast, our method yields more consistent predictions across both views and tasks.

Abstract

This paper addresses the challenge of training a single network to jointly perform multiple dense prediction tasks, such as segmentation and depth estimation, i.e., multi-task learning (MTL). Current approaches mainly capture cross-task relations in the 2D image space, often leading to unstructured features lacking 3D-awareness. We argue that 3D-awareness is vital for modeling cross-task correlations essential for comprehensive scene understanding. We propose to address this problem by integrating correlations across views, i.e., cost volume, as geometric consistency in the MTL network. Specifically, we introduce a lightweight Cross-view Module (CvM), shared across tasks, to exchange information across views and capture cross-view correlations, integrated with MTL encoder features for multi-task prediction. This module is architecture-agnostic and can be applied to both single- and multi-view data. Extensive results on NYUv2 and PASCAL-Context demonstrate that our method effectively injects geometric consistency into existing MTL methods to improve performance.

1. Introduction

One central focus of multi-task learning (MTL) in computer vision [61] is to jointly solve multiple visual tasks within a single network. By sharing the majority of model parameters, MTL models effectively reduce computational cost and memory capacity while exploiting cross-task inductive biases [8, 32, 61, 76]. This multi-task collaborative framework resonates deeply with a variety of real-world applications [76], such as robotic automation, where depth estimation and spatial awareness for obstacle avoidance are combined with semantic segmentation and other scene understanding tasks for object localization [1].

However, building an MTL model that performs consistently well for all desired tasks remains a challenging problem as it requires the MTL model to maintain a good balance between shared and task-specific features [32]. Modern deep learning-based MTL methods have explored various architectural innovations to address this: Liu *et al*. [38] introduce task-specific attention modules with a fully shared feature encoder for more flexible feature sharing; Vandenhende *et al*. [61] and Bruggemann *et al*. [6] de-

sign cross-task attention modules to capture inter-task relations from multi-scale features; recent transformer-based methods have advanced MTL through techniques such as high-resolution multi-scale decoding [72], prompt learning [64, 73], task-specific experts [15, 18, 70, 74], and multi-teacher knowledge distillation [23, 43, 49].

Despite advancements, most existing MTL methods predominantly rely on mapping 2D images to high-dimensional features and per-pixel supervision (as shown in Fig. 1 (a)), resulting in unstructured features. This unstructured feature space, coupled with a lack of explicit mechanisms for modeling spatial consistency, can lead to noisy inter-task relations and degraded performance [32, 82] (Fig. 1 (c) top). In response, pioneering work like 3DMTL [32] explores integrating 3D regularization into MTL by proposing a structured 3D-aware regularizer that projects shared features into a 3D feature space and decodes multiple tasks via differentiable rendering. Another approach, MuvieNeRF [82] recasts multi-task dense prediction as multi-view synthesis, embedding both cross-view and cross-task attention within a NeRF [45] backbone to synthesize multiple scene properties. However, despite utilizing multi-view data for 3D-awareness via feature projection and rendering, 3DMTL [32] does not directly extract and integrate multi-view geometric cues into its shared representations. Meanwhile, MuvieNeRF [82] requires multi-view data and camera parameters during inference, which limits its scalability for real-world MTL applications.

On the other hand, recent work on multi-view scene reconstruction, such as VGGT [62], MVSplat [12], and DepthSplat [68], demonstrates the success of leveraging multiple views for building robust 3D representations. MVSplat [12] effectively reconstructs high-fidelity 3D scenes by efficiently processing sparse multi-view images with 3D Gaussian Splatting. Complementing this, DepthSplat [68] integrates depth information from a pre-trained depth estimation model with Gaussian Splatting to achieve superior 3D reconstruction quality. VGGT [62] leverages a unified transformer architecture enhanced with visual geometry grounding to improve 3D understanding from multi-view inputs and multiple downstream 3D tasks. However, these methods primarily focus on scene reconstruction or 3D representation and are not directly designed to enhance multi-task learning that jointly performs multiple dense prediction tasks from single view image input, including depth estimation, boundary detection, surface normal estimation, and semantic segmentation, making their integration into MTL unclear. This leads us to a critical question for current MTL frameworks:

Can we introduce cross-view correlations into MTL to ensure high geometric consistency across vision tasks?

Motivated by this insight, we propose a 3D-aware multi-task learning framework that tightly integrates the design principles of MTL and 3D reconstruction, as shown in Fig. 1 (b). Our approach augments the monocular multi-task learning feature encoder with multi-view geometric cues via a dedicated geometric pathway, which we name the Cross-view Module (CvM), allowing the model to learn representations that are simultaneously task-aligned and geometry-aware (Fig. 1 (c)). The CvM consists of three sequential components: *(i)* a spatial-aware encoder for extracting geometric and spatial primitives, *(ii)* a multi-view transformer that ingests these encoded features for relational interactions, and *(iii)* a cost volume module that reconstructs cross-view correlations as a geometric representation. Crucially, the spatial-aware encoder operates independently of the main monocular MTL pathway, allowing it to leverage strong inductive biases for spatial locality to explicitly capture geometry-rich cues. Its features are then passed into the multi-view transformer, where intra- and cross-view attention forge robust geometry-aware representations, culminating in the construction of a cost volume that establishes dense correspondence across views. These 3D-aware features complement the original monocular MTL features and are concatenated with them before being passed into lightweight, task-specific decoder heads. Our method supports both MTL training on multi-view data (or video inputs) and single-view, while only a single image is needed for inference, making it broadly applicable in practice.

To summarize, our main contributions are as follows:

- Unlike prior work that primarily focuses on learning direct mappings between input images and desired task ground-truths, we propose to enable 3D-aware MTL by integrating cross-view correlation, *i.e.*, cost volume, as a geometric consistency into multi-task learning through an introduced multi-view module.
- Our method is architecture-agnostic, allowing seamless integration into various existing MTL architectures to enhance their performance.
- Our approach is applicable to both single- and multi-view data during training, yet requires only a single image for inference, eliminating the need for camera parameters at deployment.
- Extensive experimental results demonstrate the effectiveness and superiority of our proposed 3D-aware multi-task learning framework on standard NYUv2 and PASCAL-Context benchmarks.

2. Related Work

2.1. Multi-task Learning

Multi-task Learning (MTL) [8] aims at learning a single network that jointly estimates accurate predictions for multiple desired tasks. Recent research on multi-task learning in computer vision can be broadly divided into two

categories [50, 61]. The first group aims at addressing the unbalanced optimization issues - each task's loss function often exhibits distinct scales and convergence behaviors, jointly minimizing them can lead to optimization conflicts and performance degradation. To address this issue, prior work proposes to estimate loss weights dynamically [13, 20, 25, 35, 37, 38, 53], mitigating conflicts between gradient conflicts [14, 16, 36, 58, 77], or aligning features with single-task models [30, 31].

Our work is more related to the second one which aims at designing architectures [3, 4, 6, 27, 33, 51, 57, 59, 65, 80, 81] that better share information across tasks using cross-task attention mechanisms [46], task-specific attention modules [2, 38], cross-task feature interaction [60, 72], gating strategies or mixture of experts modules [5, 15, 18, 21, 74], visual prompting [39, 73] and distillation of multiple visual foundation models [23, 43, 49]. However, these methods mainly capture cross-task relations within the 2D space, and two recent works [32, 82] show that 3D-awareness is vital for learning more structured features that are shared and beneficial for all tasks by using NeRF as a decoder or a 3D-aware regularizer. However, MuvieNeRF [82] requires multiple views and ground-truth camera parameters, limiting its use for practical scenarios. While 3DMTL [32] does not require camera parameters during inference and can be applied to standard MTL settings with single-view input, it is shown that the method has limited capacity of leveraging multi-view data for enhancing the 3D-awareness.

Unlike existing methods, we propose to enable MTL to be 3D-aware by equipping the standard MTL encoder with a cross-view module capturing cross-view relations, allowing the model to learn representations that are simultaneously task-aligned and geometry-aware. Additionally, our method is architecture agnostic and can be applied to both single and multi-view data without requiring camera parameters during inference.

2.2. 3D Scene Reconstruction and Synthesis

Our approach is also related to methods that learn 3D scene representations for multi-view scene reconstruction and synthesis [7, 9, 12, 19, 29, 41, 45, 62, 63, 69, 79]. Earlier works [26, 45] in this field represent only a single scene per model, require many calibrated views, or are not able to perform tasks other than novel view synthesis such as semantic segmentation, depth estimation. Zhi *et al*. [83] extend the standard NeRF pipeline through a parallel semantic segmentation branch to jointly encode semantic information of the 3D scene, and obtain 2D segmentations by rendering the scene for a given view. Panoptic Neural Fields [28] predict a radiance field encoding color, density, instance, and category labels for any 3D point by combining multiple encoders for both background and individual object instances. However, this approach is limited to predicting these tasks on novel views of previously seen scenes. Consequently, it cannot be applied to entirely new scenes without additional training and is constrained to handling only rigid objects.

PixelNeRF [75] and PixelSplat [10] condition a NeRF [45] or Gaussian Splatting [26] on image inputs through an encoder, allowing for the modeling of multiple scenes jointly and generalizing to unseen scenes, however, the work focuses only on synthesizing novel views. MVSplat [12] further improves PixelSplat [10] by efficiently incorporating cross-view correlations to improve the scene reconstruction. Building on this, DepthSplat [68] proposes to leverage depth prediction from a single-view depth model for further improving the quality of 3D scene reconstruction. More recently, Wang *et al*. [62] propose VGGT that directly infers a set of 3D scene attributes from multiple views using a single, efficient feed-forward transformer.

In contrast to these methods that focus on scene reconstruction or synthesis, our work focuses on joint learning of dense vision problems in novel scenes and leverages cross-view correlation as a geometry cue to bring a beneficial structure to the learned representations. Our method can be trained from single-view or multi-view inputs and is not limited to a fixed architecture or specific set of tasks.

3. Methodology

3.1. Multi-task Learning

In multi-task learning (MTL), we aim to train a single model that takes an RGB image as input and simultaneously predicts multiple dense output targets, such as depth, edges, semantic labels, and surface normals. Formally, with an input image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$, the goal is to estimate a set of task-specific outputs $\mathcal{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_T\}$ corresponding to T different tasks.

A common approach to MTL is to employ a shared encoder with T task-specific decoders architecture, where a feature extractor $f(\cdot)$ maps the input image to a latent representation $f(\boldsymbol{I}) \in \mathbb{R}^{H'\times W'\times C}$. This shared representation is then processed by T lightweight task-specific decoders $\{h_t\}_{t=1}^{T}$ to generate the task predictions $\hat{\boldsymbol{y}}_t = h_t \circ f(\boldsymbol{I})$. Such designs exploit the redundancy between related tasks and improve training efficiency by sharing features across tasks.

The model is typically trained on a single-view labeled dataset $\mathcal{D}$ with N image-label pairs by jointly minimizing multiple losses:

$$\min_{f,\{h_t\}_{t=1}^{T}} \frac{1}{N} \sum_{(\boldsymbol{I},\mathcal{Y})\in\mathcal{D}} \sum_{\boldsymbol{y}_t\in\mathcal{Y}} \ell_t(h_t \circ f(\boldsymbol{I}), \boldsymbol{y}_t), \qquad (1)$$

where ℓ_t denotes the task-specific loss function, e.g., cross-entropy for segmentation, L_1 loss for depth estimation.

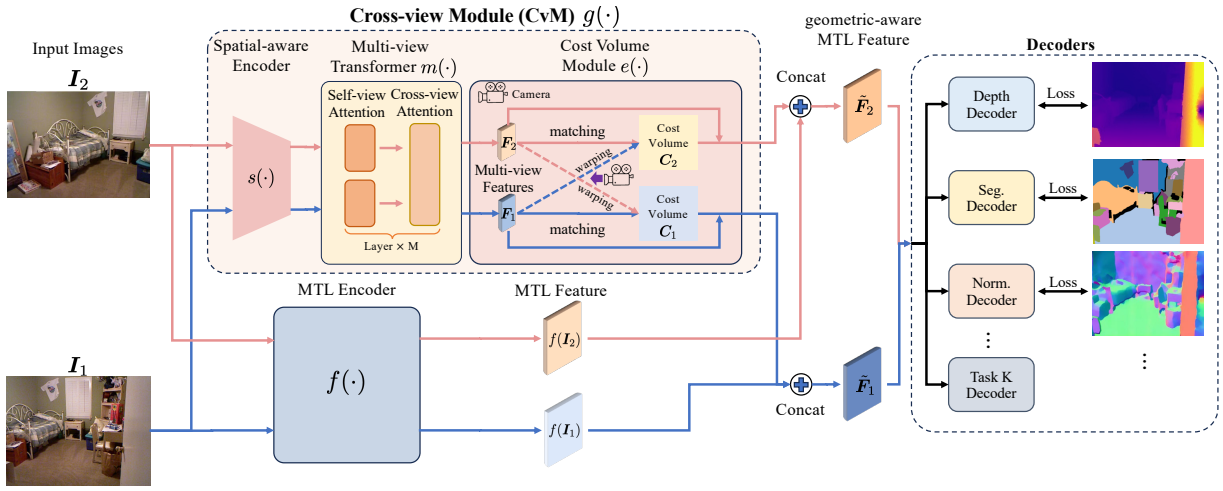


Figure 2. **Illustration of our method for integrating cross-view correlation for enabling 3D-aware MTL.** Given an image $\boldsymbol{I}_1$, we feed it and its neighboring view $\boldsymbol{I}_2$ into the MTL encoder $f(\cdot)$ and extract the MTL features $f(\boldsymbol{I}_1)$ and $f(\boldsymbol{I}_2)$. In parallel, our lightweight cross-view module (CvM) $g(\cdot)$ takes as input both views. In CvM, a spatial-aware encoder $s(\cdot)$ encodes geometrically biased features, followed by a multi-view transformer $m(\cdot)$ that enables information exchange across views and outputs cross-view features $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$. A cost volume module $e(\cdot)$ then converts $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ to the cost volume $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ by warping the feature from one view to another given their relative camera parameters and matching features across views. Finally, both cost volume and cross-view feature are concatenated with the MTL features, forming the geometric-aware MTL feature $\tilde{\boldsymbol{F}}_1$ and $\tilde{\boldsymbol{F}}_2$ for estimating predictions of multiple dense vision tasks.

3.2. MTL with Cross-view Correlations

While existing multi-task learning benefits from feature sharing, it relies on single-view 2D images, and *its inherent lack of 3D awareness often results in geometric inconsistencies between related tasks*. To mitigate this, one straightforward approach is to include multi-view data (e.g., video sequences) for training MTL models to improve geometric consistency. However, simply mapping multi-view inputs to the desired task ground-truths does not ensure consistency across views of the same scene, which is a crucial geometric cue for scene understanding.

To address this, we propose a 3D-aware multi-task learning framework that augments the shared encoder $f(\cdot)$ with a lightweight cross-view module (CvM) $g(\cdot)$—*shared across all tasks*—comprising: (i) a spatial-aware encoder that extracts geometry-biased features from paired views; (ii) a multi-view transformer that performs self/cross-attention to exchange information and produce cross-view enhanced features; and (iii) a differentiable cost volume builder that warps and matches features across depth hypotheses to explicitly encode cross-view correlations. The resulting geometric representation is fused with features from $f(\cdot)$ to obtain task-specific predictions (Fig. 2), enforcing cross-view geometric consistency and improving 3D coherence across tasks. Fig. 2 illustrates the overall pipeline of our method. The detailed design of our CvM module is as follows:

Spatial-aware encoder provides geometry-biased features from each view that are decoupled from monocular MTL cues, to serve as clean inputs for cross-view correlation modeling shared by all tasks. More specifically, given an image $\boldsymbol{I}_1$, we feed it and its neighboring views $\{\boldsymbol{I}_i\}_{i=2}^{V}$ into the MTL encoder to extract the MTL features $\{f(\boldsymbol{I}_i)\}_{i=1}^{V}$. In this work, we use 1 neighboring view, *i.e.*, $V = 2$, but the method supports more views. One could simply utilize the MTL features for cross-view matching and regularizing cross-view consistency. However, we argue that this can lead to interference between monocular MTL and cross-view matching, leading to higher difficulty in training (shown in Tab. 5) and extending to other MTL models.

To this end, we instead design a cross-view module that operates in parallel with the MTL encoder. This cross-view module consists of a spatial-aware encoder $s(\cdot)$ that extracts geometric-aware features, followed by a multi-view transformer $m(\cdot)$ which aggregates intra- and cross-view correspondences, and a cost volume module $e(\cdot)$ for constructing cross-view correlations as geometric representations. The spatial-aware encoder $s(\cdot)$ is implemented as a shallow ResNet-like [22] Convolutional Neural Network (CNN), similar to [12, 67], to extract spatial-aware downsampled features of all views $\{s(\boldsymbol{I}_i)\}_{i=1}^{V}$.

Multi-view transformer aggregates complementary intra- and cross-view context to strengthen correspondences and disambiguate occlusions/textureless regions, yielding cross-view enhanced features for subsequent geometry construction. Instead of directly matching spatial-aware features $\{s(\boldsymbol{I}_i)\}_{i=1}^{V}$, we adopt a multi-view transformer, implemented as a multi-view Swin Transformer [40, 66–68], consisting of stacked self- and cross-attention layers to facilitate information exchange

across views. Within this transformer, for each view, we compute attention with respect to its neighboring views[1], enabling the transformer to aggregate complementary visual cues and disambiguate challenging regions such as occlusions and textureless surfaces. To balance computational efficiency and geometric consistency, we follow a local attention design similar to Swin Transformer [40], where attention is restricted within spatial windows but is repeated across all scales and views. This ensures that the resulting features are both geometry-aware and scalable to large input resolutions, which is important for dense vision tasks. The output of the multi-view transformer is a set of cross-view enhanced feature maps $\{\boldsymbol{F}_i\}_{i=1}^{V} = m(\{s(\boldsymbol{I}_i)\}_{i=1}^{V})$, which are subsequently used for constructing the cost volume.

Cost volume module converts learned correspondences into an explicit, differentiable 3D representation by building a depth-parameterized cost volume that enforces geometric consistency. Given cross-view enhanced feature maps $\{\boldsymbol{F}_i\}_{i=1}^{V}$, we aim to encode the feature matching information across views as a geometric cue, shared across all dense vision tasks, to improve their performance. Following prior work in multi-view stereo [67, 71], we construct a differentiable cost volume to explicitly model the geometric consistency across views. To achieve this, we first define a set of L candidate depth planes $\{d_1, d_2, \ldots, d_L\}$ sampled uniformly in inverse depth space. For each candidate depth d, the feature of one neighboring view $\boldsymbol{I}_j$ is warped to the reference view $\boldsymbol{I}_i$ using their camera intrinsics and relative pose, producing $\hat{\boldsymbol{F}}_{j\to i}^{(d)}$.

For each view $\boldsymbol{I}_i$, we perform pixel-wise matching between its feature and the warped features from each neighboring view under each depth candidate using dot-product similarity, and aggregate the resulting matching scores over all neighboring views:

$$\boldsymbol{C}_i^{(d)} = e(\{\boldsymbol{F}_i\}_{i=1}^{V}) = \frac{1}{V-1}\sum_{\substack{j=1\\ j\neq i}}^{V} \frac{\boldsymbol{F}_i \cdot \hat{\boldsymbol{F}}_{j\to i}^{(d)}}{\sqrt{K}}, \quad (2)$$

where K is the channel dimension for normalization. This yields a 3D cost volume $\boldsymbol{C}_i \in \mathbb{R}^{H\times W\times L}$ for each view shared across all tasks.

Training objective. Finally, we concatenate the cost volume $\boldsymbol{C}_i$ and the cross-view enhanced feature $\boldsymbol{F}_i$ with the MTL feature $f(\boldsymbol{I}_i)$ to form the 3D-aware multi-task feature $\tilde{\boldsymbol{F}}_{\boldsymbol{I}_i} = \text{concat}\left(f\left(\boldsymbol{I}_i\right), \boldsymbol{C}_i, \boldsymbol{F}_i\right)$ for estimating multi-task predictions. We then measure the mismatch between the ground-truth labels and the predictions obtained from the spatial-aware MTL feature, and jointly optimize the model by minimizing all task losses as in Eq. (1):

$$\min_{f,\{h_t\}_{t=1}^{T},g} \frac{1}{NV} \sum_{\{(\boldsymbol{I}_i,\mathcal{Y}_i)\}_{i=1}^{V}\in\mathcal{D}} \sum_{\boldsymbol{y}_t\in\mathcal{Y}_i} \ell_t(h_t(\tilde{\boldsymbol{F}}_{\boldsymbol{I}_i}), \boldsymbol{y}_t), \quad (3)$$

where $g = e\circ m\circ s$ is the cross-view module that consists of the spatial-aware encoder s, multi-view transformer m and the cost volume module e.

Training and inference with single-view inputs. Although our cross-view module requires at least two views as input, during inference, often only a single-view image is available. We address this by duplicating the single-view image to serve as the neighboring view, enabling multi-task predictions. We employ the same strategy for training our method on single-view datasets and found that it still performs effectively (as shown in Tab. 2 and Tab. 3). We hypothesize that training the cross-view module on duplicated single-view images can prevent it from capturing spurious correlations between identical views, thereby enhancing its robustness. However, further improvements could be achieved through augmentation or multi-view image generation techniques, and we leave this for future work.

4. Experiments

In this section, we first detail the benchmarks and our implementation, followed by a quantitative and qualitative analysis of our method. Please refer to the supplementary materials for more results and details.

4.1. Datasets

NYUv2 [55] is a popular MTL benchmark consisting of 1449 indoor RGB-D images captured with a Microsoft Kinect sensor. We follow the standard split [17] and we use 795 and 654 images for training and testing. Following the prior work [43, 72], we perform four tasks, including 40-class semantic segmentation, depth estimation, surface normal prediction, and boundary detection.

NYUv2 Video Frames. Following prior work [32], we leverage raw RGB-D video sequences from the NYUv2 dataset [55], extracting additional video frames to construct multi-view inputs. We also follow 3DMTL [32] and use COLMAP [52] to estimate relative camera poses between adjacent frames. These video frames only have depth annotations and they are used for training in multi-view setting.

PASCAL-Context [11] provides dense annotations for various visual tasks, including semantic segmentation, boundary detection, and human part segmentation. We follow [61] and also perform saliency detection, and surface normal prediction with annotation from Vandenhende *et al*. [61]. We adopt the standard splits [61, 72]: 4998 images for training and 5105 images for testing.

[1]Our method supports $V > 2$ though we use $V = 2$ by default. For each reference view that has more than 2 neighboring views (*i.e.*, $V > 3$), we perform cross-attention between the reference view and its top-2 nearest neighboring views, which are selected based on their camera distances to the reference view to ensure better trade-off between performance and computational efficiency.

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
SAK [43] *w/o video*	**63.18**	0.4313	16.25	79.43	0.00
SAK [43]	62.60	0.4093	16.19	79.58	1.19
Ours	62.78	**0.4034**	**16.10**	**80.52**	**2.03**
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
3DMTL*	64.25	0.3952	**15.24**	80.15	1.68
Ours	**65.27**	**0.3836**	15.35	**81.69**	**3.09**

Table 1. Quantitative comparison of our method 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.

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
STL	54.19	0.5560	19.22	78.09	0.00
MTL	52.42	0.5413	19.29	76.50	-0.76
TaskExperts [74]	55.35	0.5157	18.54	78.40	3.33
BFCI [78]	55.51	0.4930	18.47	78.22	4.46
TSP [64]	55.39	0.4961	18.44	77.50	4.07
MLoRE [70]	55.96	0.5076	18.33	78.43	4.26
InvPT [72]	53.56	0.5183	19.04	78.10	1.64
3DMTL [32]	54.87	0.5006	18.55	78.30	3.74
RADIO [49]	59.32	0.4698	17.46	79.41	8.95
Ours	**60.26**	**0.4619**	**17.34**	**80.36**	**10.20**
SAK [43]	**63.18**	0.4313	16.25	79.43	14.05
Ours	63.12	**0.4044**	**16.22**	**80.56**	**15.63**
DINOv3 [56]	63.68	0.4113	15.53	80.10	16.33
Ours	**64.98**	**0.3909**	**15.27**	**81.58**	**18.66**

Table 2. Quantitative comparison of our method to the SotA methods on NYUv2 dataset. ΔMTL is computed using single-task learning "STL" as baseline.

4.2. Implementation Details

Our method is architecture agnostic and can be applied to different state-of-the-art MTL methods. We apply our method to the recent 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 the same loss functions and loss weights as in [43, 61, 72]. Across all experiments, we use ViT-L as the backbone for MTL encoder. For CvM, our spatial-aware encoder produces 128-dimensional features at $1/8$ input resolution, followed by the multi-view transformer with 6 self- and cross-view attention layers. The number of depth candidates L is set to be 128 to uniformly sample depth candidates from 0.0001 to 10. Please refer to supplementary for more details.

Evaluation Metrics. We follow the previous methods [43, 72], measuring semantic segmentation and human parsing by the mean Intersection over Union (mIoU), saliency detection by maximum F-measure (maxF), surface normal estimation by mean error (mErr) of angles, depth estimation by Root Mean Square Error (RMSE), and boundary detection by optimal-dataset scale F-measure (odsF) [44, 48]. We also report the multi-task learning performance ΔMTL, *i.e*., average performance gains across all tasks w.r.t. a baseline, *e.g*., single-task learning method, as in prior work [61].

4.3. Results

MTL with Multiple Views. We perform experiments on multi-view data by training models on both single-view training images and video frames on NYUv2 and evaluating models on the single-view testing set of NYUv2. We incorporate our method with SAK [43] and DINOv3 [56] in this setting, and the results are depicted in Tab. 1. We observe that for both SAK [43] and DINOv3 [56], simply introducing multi-view data for training improves MTL performance, such as depth and surface normal estimation. However, this does not fully exploit the cross-view correlations. In contrast, by explicitly modeling spatial correspondence and aggregating cross-view features, our CvM enables more effective use of multi-view signals. Notably, DINOv3 with CvM trained on multi-view data achieves +1.57 over DINOv3 trained with multi-view data, and +3.09 over the DINOv3 baseline trained with single-view data.

We further compare our approach with 3DMTL [32], which injects 3D awareness into MTL through a neural rendering regularization. Our CvM achieves consistently better results, improving segmentation by +1.0, boundary detection by +1.5, and reducing depth RMSE from 0.3952 to 0.3836, while achieving comparable performance on surface normal. These results strongly indicate that the cross-view correlations learned by our method effectively enhance 3D awareness and hence improve MTL framework.

Comparisons with SotAs. Our method is not limited to training on multi-view input and can be applied in a single–view setting for comparison with current state-of-the-art (SotA) MTL methods. We compare our method with SotA methods and report the results on NYUv2 and PASCAL in Tab. 2 and Tab. 3, respectively.

On NYUv2, integrating our method with state-of-the-art MTL methods consistently improves their MTL performance by an average over 1.7. Notably, our approach leads to comprehensive improvements on RADIO [49] and DINOv3 [56] across all tasks, and surpasses SAK [43] in three out of four tasks, with comparable segmentation result. Moreover, our CvM demonstrates clear benefits in geometry-intensive tasks such as depth estimation. Across three MTL methods, our method improves depth by 4.29% on average (*e.g*., from 0.4113 (DINOv3) to 0.3909 (Ours)) and boosts boundary detection F-score by 1.2 on average (*e.g*., Ours vs DINOv3: 81.58 vs 80.10). Our method with DINOv3 achieves a new SotA across all tasks on NYUv2.

On PASCAL-Context, our method also brings consistent

Method	Seg. (mIoU) ↑	PartSeg (mIoU) ↑	Sal (maxF) ↑	Normal (mErr) ↓	Boundary (odsF) ↑	ΔMTL ↑
STL	81.61	72.77	83.80	13.87	75.24	0.00
MTL	79.26	68.28	84.16	14.06	71.59	-2.97
TaskExpert [74]	80.64	69.42	84.87	13.56	73.30	-0.97
BFCI [78]	80.64	70.06	84.64	13.82	72.96	-1.32
TSP [64]	81.48	70.64	84.86	13.69	74.80	-0.22
MLoRE [70]	81.41	70.52	84.90	13.51	75.42	0.16
InvPT [72]	79.03	67.61	84.81	14.15	73.00	-2.81
3DMTL [32]	79.53	69.12	84.94	13.53	74.00	-1.08
RADIO [49]	81.11	71.50	85.17	13.49	74.80	0.29
Ours	**81.18**	**71.75**	**85.26**	**13.42**	**76.95**	**1.07**
SAK [43]	84.01	76.99	84.65	13.82	76.27	2.30
Ours	**84.41**	**77.68**	**84.83**	**13.61**	**76.79**	**3.07**
DINOv3 [56]	84.07	77.29	84.40	13.70	76.30	2.52
Ours	**84.56**	**77.97**	**84.56**	**13.66**	**79.29**	**3.71**

Table 3. Quantitative comparison of our method to the SotA methods on PASCAL-Context dataset. ΔMTL is computed using single-task learning "STL" as baseline.

improvements over SotA MTL approaches. By integrating CvM into RADIO [49], SAK [43], and DINOv3 [56], our method yields gains across all tasks and boosts MTL performance. The results show that this hybrid design is beneficial: despite the absence of multi-view inputs, our CvM still helps to prevent MTL encoder from capturing noisy and spurious view correlations between identical views and further improves the MTL performance.

Training Cost. Our CvM introduces approximately 5M additional parameters, making it a lightweight component relative to the overall size of typical MTL encoders, *e.g.*, MTL models like RADIO [49], SAK [43], and DINOv3 [56] typically contain 300–350M parameters. Our proposed CvM accounts for only 1.5% of the total parameter count of MTL encoder.

4.4. Ablation Study

Here we conduct an in-depth analysis of our method to validate the effectiveness of our CvM. All experimental analyses are performed on NYUv2 dataset. Video frames in NYUv2 are used during training. Please refer to supplementary for more detailed results of Tabs. 5 to 7.

Cost Volume & Cross-view Features. We aim to examine the contributions of the cost volume $\boldsymbol{C}_i$ and the cross-view enhanced features $\boldsymbol{F}_i$. We start with our method without both $\boldsymbol{C}_i$ and $\boldsymbol{F}_i$, resulting in a standard MTL baseline "Ours *w/o CV & CF*". We then add only the cost volume, *i.e.*, "Ours *w/o CF*" to verify the effectiveness of cost volume. Based on "Ours w/o CF", we further add the cross-view features, which is our full model "Ours".

The results in Tab. 4 show that the cost volume alone can effectively supplement the MTL encoder with cross-view geometric cues, improving the MTL performance (ΔMTL) by over 1% on average. When combined with the cross-view enhanced features, our model achieves a further boost in performance, indicating that the two components are complementary. These findings validate the effectiveness of our design and demonstrate that injecting cross-view information into a standard MTL encoder is beneficial for learning geometry-aware representations that enhance performance across multiple tasks.

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
Ours	**65.27**	**0.3836**	15.35	**81.69**	**19.05**

Table 4. Ablation study on cross-view module. "Ours *w/o CV & CF*" is the baseline without our cross-view module. "Ours *w/o CF*" indicates our method that only uses cost volume. "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.

Extracting spatial-aware features plays a crucial role for subsequent cost volume construction and 3D-aware MTL. Apart from our design, we also consider three other methods and report results in Tab. 5: (1) "MTL encoder" uses the feature from the MTL encoder as spatial-aware feature. (2) "MTL encoder + LoRA [24]" attaches low-rank adapters (LoRA) [24] (rank and α are set to 16, scaling factor s=1) into the MTL encoder to adapt the features from MTL encoder as spatial-aware features. (3) "MTL encoder + Adapter [43]" appends adapters from SAK [43] to the MTL encoder for encoding spatial-aware features. As shown in Tab. 5, our design achieves the highest MTL performance. We attribute this to the strong inductive bias of CNNs in modeling spatial structures, resulting in higher-quality spatial-aware features. Beyond performance advantages, using a lightweight independent spatial-aware encoder instead of modifying the MTL encoder offers a non-intrusive mechanism to supply 3D-aware features, making our method architecture-agnostic and easily integrable with various MTL backbones.

Number of Depth Candidates L can affect the reconstruction of cost volume, and we experiment with 128, 256, 384, and 512 candidates while keeping the depth range fixed to investigate the effect of L. As shown in Tab. 6, increasing the number of candidates to 512 improves performance. However, using 512 depth candidates inevitably increases the computational cost significantly. For better trade-off between efficiency and performance, we use 128 depth candidates. This also aligns with the choices in 3D reconstruction pipelines such as MVSplat [12] and DepthSplat [68].

Number of Views. We use one neighboring view (V=2) while our method supports more views. We performed experiments with $V = 2, 3, 4$ and results are reported in Tab. 7. We can see that increasing the number of views can help learn better geometric structures and improve performance, while using $V = 2$ is sufficient and efficient.

Spatial-aware Encoder	MTL Encoder	MTL Encoder + LoRA [24]	MTL Encoder + Adapter [43]	Ours
ΔMTL ↑	18.84	18.87	18.98	**19.05**

Table 5. Comparisons of various spatial-aware feature extraction methods in cross-view module on NYUv2. ΔMTL is computed using "STL" in Tab. 2 as baseline.

Number of Depth	96	128	256	384	512
ΔMTL ↑	18.88	19.05	18.62	18.46	**19.25**

Table 6. Comparisons of various number of depth candidates on NYUv2. ΔMTL is computed using "STL" in Tab. 2 as baseline.

Number of Views	4	3	2
ΔMTL ↑	18.63	**19.20**	19.05

Table 7. Results of various number of views. ΔMTL is computed w.r.t. "STL" in Tab. 2.

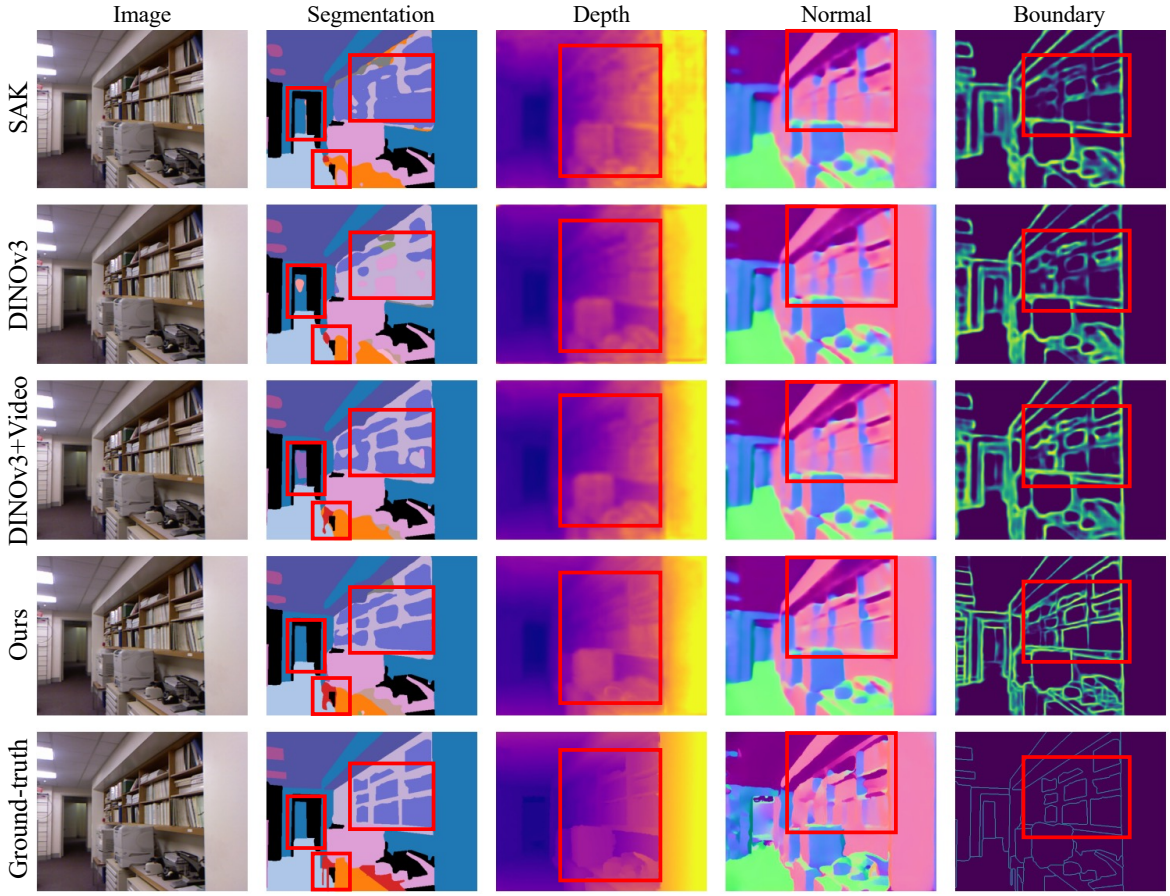


Figure 3. **Qualitative Comparisons on NYUv2.** The first column shows the RGB image, and the remaining columns display either the ground-truth or model predictions. The last row shows the ground-truth of four tasks. The first to fourth rows show the predictions of SAK, DINOv3, DINOv3 trained with videos as multi-view data, and our method, respectively.

4.5. Qualitative Results

We visualize task predictions for four methods: SAK [43], DINOv3 [56], DINOv3 trained with multi-view video data, and Ours. As shown in Fig. 3, SAK and single-view DINOv3 mis-segment the bookshelf and table-leg areas and produce blurred depth and noisy normals. DINOv3 trained with multi-view data improves the prediction but it still fails to recover fine geometry and boundaries. Our method can be observed to estimate more accurate predictions, yielding accurate segmentation of thin structures, sharper depth discontinuities, more stable normals, and clearer boundaries. These results strongly verify that geometric information is crucial for comprehensive scene understanding, and our method is capable of injecting geometric consistency into MTL methods.

5. Conclusion and Future Work

In this paper, we demonstrate that cross-view consistency provides crucial geometric cues for multi-task dense prediction in scene understanding across several benchmarks. We introduce a Cross-view Module (CvM) for MTL that estimates cross-view correlations through a spatial-aware encoder with a multi-view transformer and cost-volume construction. Through extensive experiments, we have demonstrated that CvM can be seamlessly integrated into existing multi-task learning methods and supports both single- and multi-view input. Despite its effectiveness, our method has **limitations**: it is designed for static scenes, whereas dynamic environments with moving objects or camera motion introduce additional challenges. Future work will explore more efficient multi-view modeling and motion-aware extensions to better handle dynamic scenes.

References

[1] Stefan Ainetter, Christoph Böhm, Rohit Dhakate, Stephan Weiss, and Friedrich Fraundorfer. Depth-aware object segmentation and grasp detection for robotic picking tasks. *arXiv preprint arXiv:2111.11114*, 2021. 1

[2] Deblina Bhattacharjee, Sabine Süsstrunk, and Mathieu Salzmann. Vision transformer adapters for generalizable multitask learning. *arXiv preprint arXiv:2308.12372*, 2023. 3

[3] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *NeurIPS*, pages 235–243, 2016. 3

[4] Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C Alexander, and Jorge Cardoso. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *ICCV*, pages 1385–1394, 2019. 3

[5] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. *arXiv preprint arXiv:2008.10292*, 2020. 3

[6] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021. 1, 3

[7] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *CVPR*, pages 3981–3990, 2022. 3

[8] Rich Caruana. Multitask learning. *Machine learning*, 28(1): 41–75, 1997. 1, 2

[9] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pages 5799–5809, 2021. 3

[10] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, pages 19457–19467, 2024. 3

[11] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014. 5

[12] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, pages 370–386. Springer, 2024. 2, 3, 4, 7

[13] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 794–803. PMLR, 2018. 3

[14] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *NeurIPS*, 2020. 3

[15] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *CVPR*, pages 11828–11837, 2023. 2, 3

[16] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir A Rawashdeh. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *CVPR Workshop*, 2019. 3

[17] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015. 5

[18] Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *NeurIPS*, 35:28441–28457, 2022. 2, 3

[19] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 3

[20] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *ECCV*, pages 270–287, 2018. 3

[21] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *ICML*, pages 3854–3863. PMLR, 2020. 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 1

[23] Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. Radiov2. 5: Improved baselines for agglomerative vision foundation models. In *CVPR*, pages 22487–22497, 2025. 2, 3

[24] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 7, 8, 1

[25] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018. 3

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[27] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, pages 6129–6138, 2017. 3

[28] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *CVPR*, pages 12871–12881, 2022. 3

[29] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, pages 71–91. Springer, 2024. 3

[30] Wei-Hong Li and Hakan Bilen. Knowledge distillation for multi-task learning. In *ECCV Workshop on Imbalance Problems in Computer Vision*, pages 163–176. Springer, 2020. 3

[31] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal representations: A unified look at multiple task and domain learning. *arXiv preprint arXiv:2204.02744*, 2022. 3

[32] Wei-Hong Li, Steven McDonagh, Ales Leonardis, and Hakan Bilen. Multi-task learning with 3d-aware regularization. In *ICLR*, 2024. 1, 2, 3, 5, 6, 7, 4

[33] Jason Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 466–473, 2018. 3

[34] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 3

[35] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *NeurIPS*, 32: 12060–12070, 2019. 3

[36] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *NeurIPS*, 2021. 3

[37] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *ICLR*, 2021. 3

[38] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019. 1, 3

[39] Yajing Liu, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. Hierarchical prompt learning for multi-task learning. In *CVPR*, pages 10888–10898, 2023. 3

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *CVPR*, pages 10012–10022, 2021. 4, 5, 1

[41] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 2019. 3

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[43] Yuxiang Lu, Shengcao Cao, and Yu-Xiong Wang. Swiss army knife: Synergizing biases in knowledge from vision foundation models for multi-task learning. In *ICLR*, 2025. 2, 3, 5, 6, 7, 8, 1, 4

[44] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5):530–549, 2004. 6

[45] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3

[46] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016. 3

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 6, 1

[48] Jordi Pont-Tuset and Ferran Marques. Supervised evaluation of image segmentation and object proposal techniques. *TPAMI*, 38(7):1465–1478, 2015. 6

[49] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *CVPR*, pages 12490–12500, 2024. 2, 3, 6, 7, 1

[50] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 3

[51] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *AAAI*, pages 4822–4829, 2019. 3

[52] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5, 2

[53] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 2018. 3

[54] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 1

[55] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 5

[56] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 1, 6, 7, 8, 2, 3, 4

[57] Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. Many task learning with task routing. In *ICCV*, pages 1375–1384, 2019. 3

[58] Mihai Suteu and Yike Guo. Regularizing deep multi-task networks using orthogonal gradients. *arXiv preprint arXiv:1912.06844*, 2019. 3

[59] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. In *BMVC*, 2020. 3

[60] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, pages 527–543. Springer, 2020. 3

[61] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *TPAMI*, 2021. 1, 3, 5, 6

[62] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025. 2, 3

[63] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024. 3

[64] Shuo Wang, Jing Li, Zibo Zhao, Dongze Lian, Binbin Huang, Xiaomei Wang, Zhengxin Li, and Shenghua Gao.

Tsp-transformer: Task-specific prompts boosted transformer for holistic scene understanding. In *WACV*, pages 925–934, 2024. 2, 6, 7, 3

[65] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, pages 675–684, 2018. 3

[66] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, pages 8121–8130, 2022. 4

[67] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *TPAMI*, 45(11):13941–13958, 2023. 4, 5, 1

[68] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, pages 16453–16463, 2025. 2, 3, 4, 7

[69] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *CVPR*, pages 21924–21935, 2025. 3

[70] Yuqi Yang, Peng-Tao Jiang, Qibin Hou, Hao Zhang, Jinwei Chen, and Bo Li. Multi-task dense prediction via mixture of low-rank experts. In *CVPR*, pages 27927–27937, 2024. 2, 6, 7, 3

[71] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783, 2018. 5

[72] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, pages 514–530. Springer, 2022. 2, 3, 5, 6, 7, 1

[73] Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *ICLR*, 2023. 2, 3

[74] Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. In *ICCV*, pages 21828–21837, 2023. 2, 3, 6, 7

[75] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. 3

[76] Jun Yu, Yutong Dai, Xiaokang Liu, Jin Huang, Yishan Shen, Ke Zhang, Rong Zhou, Eashan Adhikarla, Wenxuan Ye, Yixin Liu, et al. Unleashing the power of multi-task learning: A comprehensive survey spanning traditional, deep, and pretrained foundation model eras. *arXiv preprint arXiv:2404.18961*, 2024. 1

[77] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *NeurIPS*, 2020. 3

[78] Jingdong Zhang, Jiayuan Fan, Peng Ye, Bo Zhang, Hancheng Ye, Baopu Li, Yancheng Cai, and Tao Chen. Rethinking of feature interaction for multi-task learning on dense prediction. *arXiv preprint arXiv:2312.13514*, 2023. 6, 7, 3

[79] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *CVPR*, pages 21936–21947, 2025. 3

[80] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *ECCV*, pages 235–251, 2018. 3

[81] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, pages 4106–4115, 2019. 3

[82] Shuhong Zheng, Zhipeng Bao, Martial Hebert, and Yu-Xiong Wang. Multi-task view synthesis with neural radiance fields. In *ICCV*, pages 21538–21549, 2023. 2, 3

[83] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, pages 15838–15847, 2021. 3

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.

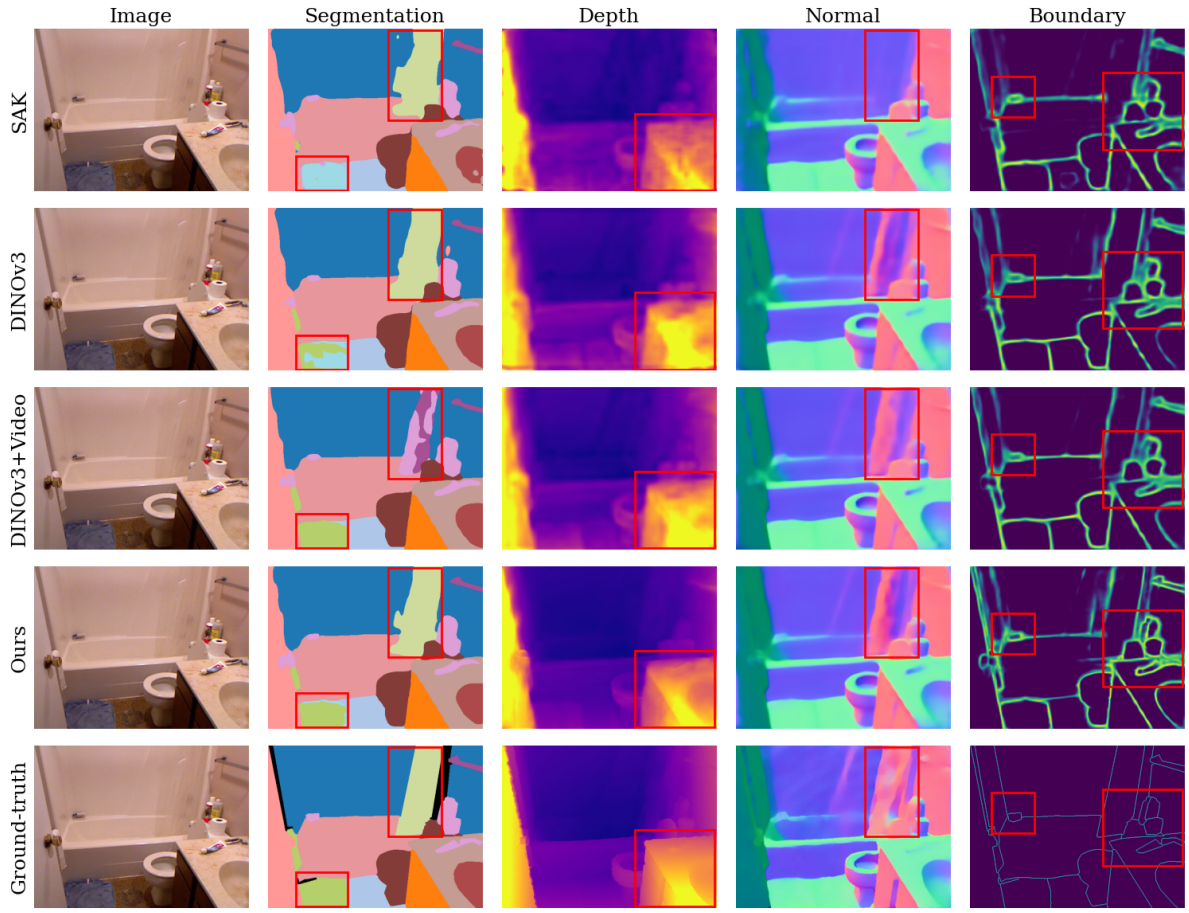


Figure A4. **Qualitative Comparisons on NYUv2.** The first column shows the RGB image, while the remaining columns present either the ground truth or model predictions. The last row shows the ground-truth of four tasks. The first to the fourth rows show the predictions of SAK, DINOv3, DINOv3 trained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

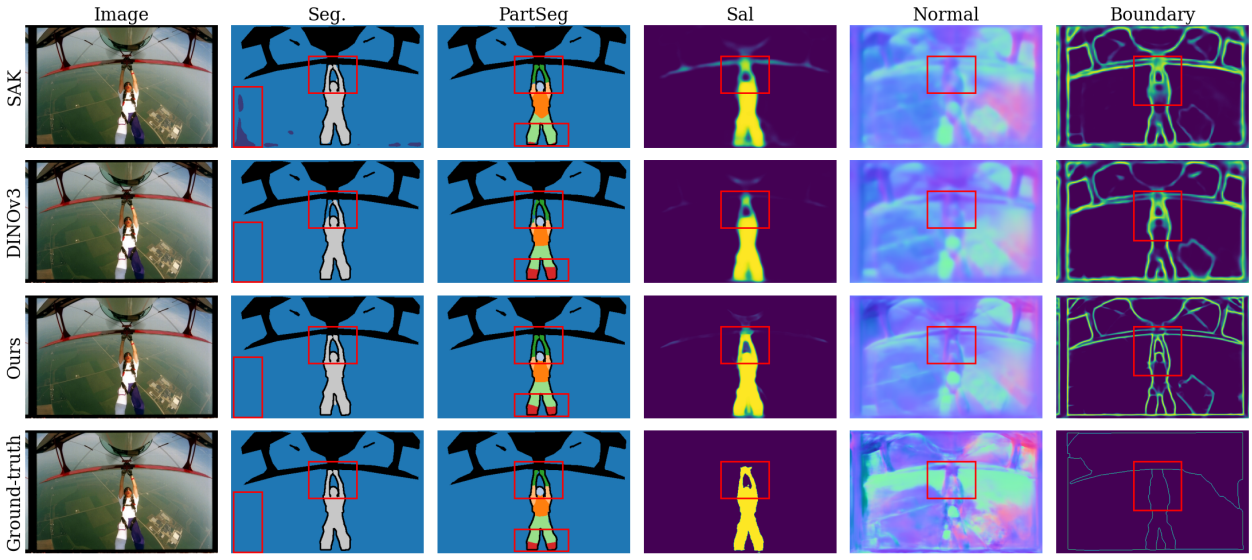


Figure A5. **Qualitative Comparisons on PASCAL-Context.** The first column shows the RGB image, while the remaining columns present either the ground truth or model predictions. The last row shows the ground-truth of five tasks. The first to the third rows show the predictions of SAK, DINOv3, and our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

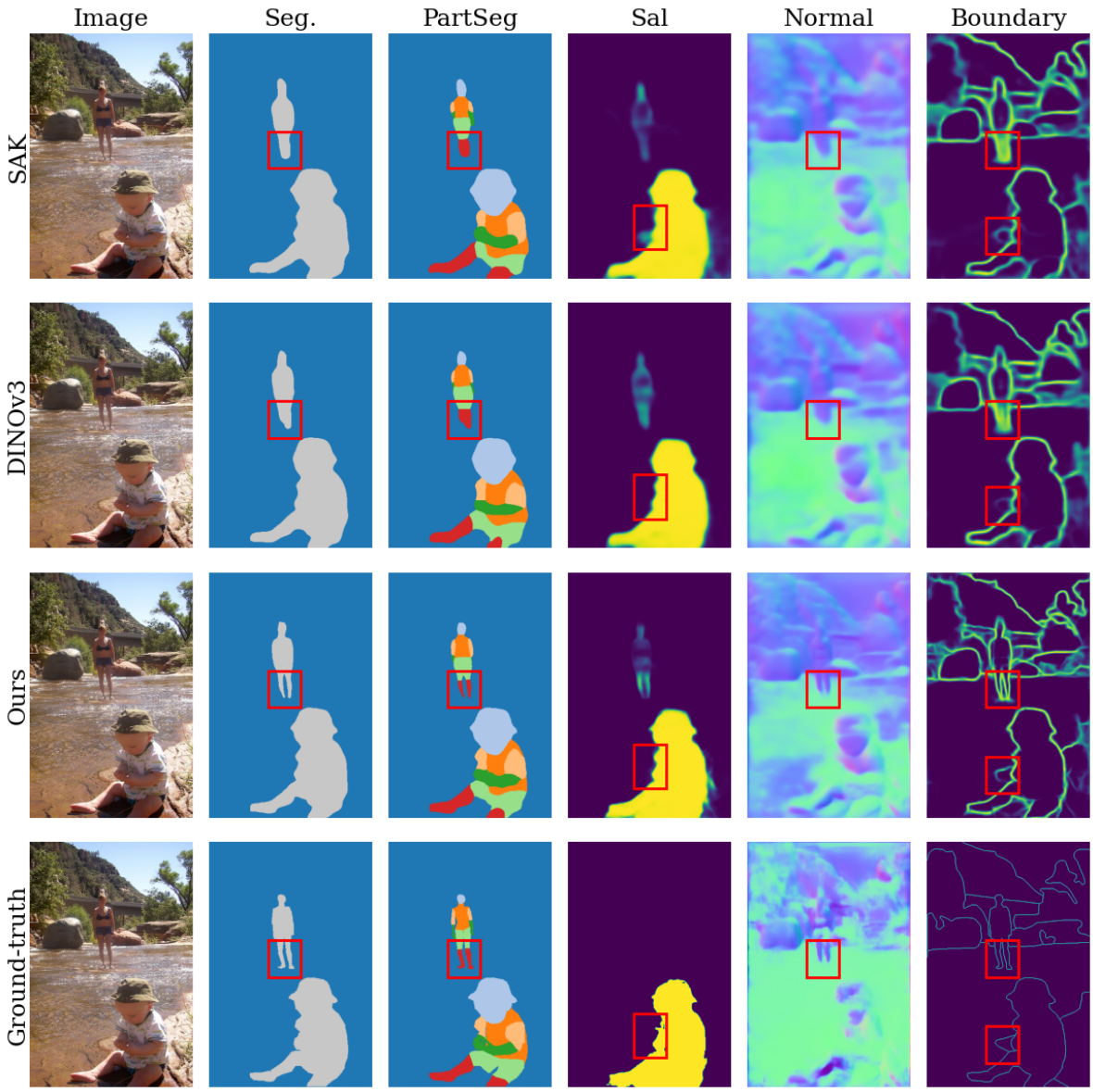


Figure A6. **Qualitative Comparisons on PASCAL-Context.** The first column shows the RGB image, while the remaining columns present either the ground truth or model predictions. The last row shows the ground-truth of five tasks. The first to the third rows show the predictions of SAK, DINOv3, and our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