Modulate-and-Map: Crossmodal Feature Mapping with Cross-View Modulation for 3D Anomaly Detection

Supplementary Material

A. Experimental Settings

A.1. Datasets and Metrics

We evaluate our method on the SiM3D benchmark [12], which consists of 8 object categories with a total of 333 instances. Each instance is captured from multiple calibrated viewpoints (12 or 36 views, depending on the object type), providing both high-resolution grayscale images (12 Mpx) and dense 3D measurements (5-7M points) that can be accessed as either point clouds or depth maps. Following the benchmark protocol, we train on a single nominal instance per object category and test on all remaining instances, which include both nominal and anomalous samples. We evaluate our method in both setups defined by SiM3D: **real-to-real**, where training uses a single real nominal instance, and **synthetic-to-real**, where training uses data rendered from a CAD model; for both setups, the testing is conducted on real data.

We adopt the evaluation metrics proposed by SiM3D. For anomaly detection, we report instance-level AUROC (I-AUROC) computed on the global anomaly scores. For anomaly segmentation, we report voxel-level AUPRO integrated up to 1% false positive rate (V-AUPRO@1%), which better reflects the stringent requirements of industrial applications compared to the more commonly used 30% threshold.

A.2. Implementation Details

Feature Extractors. For image feature extraction, we employ DINO-v2 ViT-B/14 [24], which provides features of dimension $c_I = 768$. For depth feature extraction, we use DINO-Depth, our dedicated depth encoder, trained by a self-supervised learning objective similar to DINO-v2. Since the datasets selected to train DINO-Depth contain approximately 47k samples, far from the large-scale training of DINO-v2, we train a ViT-S/14 to avert overfitting, yielding features of dimension $c_D = 384$. Both feature extractors remain frozen during training of the modulator and crossmodal mapping networks.

Network Architecture. The crossmodal mapping networks $\mathcal{M}_{I\to D}$ and $\mathcal{M}_{D\to I}$ are implemented as three-layer MLPs with hidden dimensions $[768, 576, 384]$ and $[1152, 576, 384]$, respectively. Each hidden layer is followed by GeLU activation. The view modulators Φ_I and Φ_D consist of two-layer MLPs with hidden dimension 128. Both modulators take as input the concatenated one-hot encodings of source and target views and produce scale and shift parameters for feature-wise modulation.

Training. We resize images and depth maps to 896×896 pixels and train jointly $\mathcal{M}_{I\to D}$, $\mathcal{M}_{D\to I}$, Φ_I, Φ_D for 200 epochs using the Adam optimiser [21] with an initial learning rate of 10^{-4}. We employ the OneCycleLR scheduler [31] with maximum learning rate 5×10^{-4}, cosine annealing strategy, and 10% warm-up period. During each epoch, we process all $N \times N$ source-target view pairs from the single training instance, where N is the number of views. The pairs are processed in batches of 48. This exhaustive pairing strategy ensures the network learns crossmodal mappings for all possible view relationships.

B. Additional Experiments

B.1. Computational Cost Analysis

To analyse the computational requirements of MODMAP, we compare inference times across different architectural choices on the *Sink Cabinet* class, which represents the most computationally demanding scenario in the dataset with 36 available views and unfiltered background regions. To ensure fair comparison, we measure inference times on the same machine for all architectural choices, computing the average across all the test samples from the *Sink Cabinet* class. For each sample, we record the elapsed time from data loading onto the GPU to the final computation of all the per-view anomaly maps. All time measurements are performed after GPU warm-up, and we synchronise all CUDA threads before recording the total inference time to ensure accurate timing estimates. Note that these timings exclude the volume construction step, which is identical across all methods and thus does not affect relative comparisons.

The original CFM [11] approach, using Point-MAE as the 3D backbone, requires 1654.792 ms per sample for processing all 36 views. Transitioning to the SiM3D configuration [12] with DINO-v2 features for both images and depths, increases the inference time to 3815.606 ms (2.3× overhead), primarily due to the higher-dimensional feature space for the 3D feature extractor. Despite incorporating cross-view feature aggregation, MODMAP achieves 2886.768 ms per sample – 24% faster than the CFM SiM3D configuration – while delivering superior performance through multi-view reasoning. This efficiency gain stems from our streamlined depth-based processing pipeline based on ViT-S/14[1], demonstrating that cross-view modelling can be implemented with-

[1] ViT-S/14 features 12 layers, 384 hidden size, 6 heads and 1536 MLP width, while ViT-B/14 features 12 layers, 768 hidden size, 12 heads and 3072 MLP width.

out prohibitive computational penalties. Compared to the CFM SiM3D configuration, MODMAP is advantageous in terms of both performance and cost, as it simultaneously improves detection and segmentation accuracy, while reducing inference time. Eventually, as each of the $N \times N$ anomaly maps can be individually aggregated into the anomaly volume, the runtime memory requirements pertain to N feature maps and one anomaly map at the time, thus memory usage scales as $\mathcal{O}(N)$ and not $\mathcal{O}(N \times N)$.

No. Views [#]	Inference Time [ms]	Detection	Segmentation
		Sink Cabinet	
36	2886.768	1.000	0.785
18	1443.384	1.000	0.785
9	721.692	1.000	0.783
5	400.94	0.972	0.781

Table 8. **No. Views vs. Inference Time vs. Performance** Comparison on *Sink Cabinet*, i.e., the most computationally expensive class of SiM3D.

Cost Mitigation in Low-Resource Environments. While incorporating cross-view feature mapping introduces additional computational overhead compared to single-view processing, this cost can be effectively mitigated through random view sampling without significant performance degradation. Tab. 8 demonstrates this trade-off on the *Sink Cabinet* class. Our analysis reveals that reducing the number of cross-view pairs from 36 to 18 halves inference time (from 2886.768 ms to 1443.384 ms per sample) while maintaining identical detection and segmentation performance (AUROC 1.000, AUPRO 0.785). An even more aggressive reduction to 9 views (721.692 ms) preserves perfect detection performance while yielding only a negligible 0.2% drop in segmentation quality. At 5 views, the method achieves a 7.2× speed-up over the full 36-view configuration while retaining 97.2% detection accuracy and 78.1% segmentation performance. This demonstrates that MODMAP can be adapted to resource-constrained deployment scenarios by sampling a subset of available views, offering a tunable balance between computational efficiency and anomaly detection and segmentation accuracy.

B.2. Ablations

Preliminary Study on the Image Feature Extractor. Before developing our dedicated depth encoder, we conducted a preliminary study to determine whether existing pre-trained vision models could effectively extract features from both modalities. As shown in Tab. 9, we compared using DINO-v3 and DINO-v2 as feature extractors for both images and depth maps (after treating them as single-channel images).

DINO-v2 demonstrates superior performance, achieving 0.575 I-AUROC and 0.684 V-AUPRO@1% compared to DINO-v3's 0.534 I-AUROC and 0.569 V-AUPRO@1%, representing improvements of 7.7% in detection and 20.2% in segmentation. We attribute this performance gap to differences in the training data composition. While DINO-v3's training corpus consists predominantly of social media images, which may lack industrial imagery, DINO-v2 was trained on a more diverse dataset that likely includes a broader range of visual domains.

This finding informed two key design decisions: (1) we adopted DINO-v2 as our image feature extractor, and (2) we employed the same Vision Transformer architecture[2] and training methodology for our dedicated depth encoder (DINO-Depth), ensuring architectural consistency while specialising the model for depth-based industrial anomaly detection through targeted pre-training on industrial datasets.

Features		Detection	Segmentation
Image	Depth		
DINO-v3	DINO-v3	0.534	0.569
DINO-v2	DINO-v2	**0.575**	**0.684**

Table 9. **Effects of Image Feature Extractor.**

Aggregation Function. In Sec. 3.2.4 of the main paper, we aggregate the per-view anomaly maps Ψ_I^t and Ψ_D^t by projecting them separately into 3D space and taking the maximum score at each voxel. Here, we investigate alternative strategies to aggregate the two modality maps before 3D projection. Specifically, for each view t, we compute a unified anomaly map Ψ^t by combining Ψ_I^t and Ψ_D^t using different aggregation functions $\Psi^t = \Xi(\Psi_I^t, \Psi_D^t)$, where $\Xi \in \{\max, \min, \text{prod}, \text{mean}\}$ operates element-wise on the 2D maps. The unified maps are then projected into 3D space to construct the anomaly volume.

Tab. 10 presents the results on the real-to-real setup. The maximum aggregation achieves the best performance with 0.844 I-AUROC and 0.804 V-AUPRO@1%, validating our design choice. The minimum aggregation performs poorly (0.721 I-AUROC, 0.451 V-AUPRO@1%), as it requires both modalities to detect an anomaly, significantly reducing sensitivity. This is particularly problematic for anomalies visible in only one modality (e.g., colour defects in the image or geometric defects in depth). The product and average aggregations show intermediate performance but still underperform the maximum by 9.9% and 6.2% in detection, and 16.3% and 4.8% in segmentation, respectively.

These results confirm that the maximum aggregation optimally balances the complementary information from both modalities: an anomaly is flagged if detected by either modality, maintaining high sensitivity while leveraging the preci-

[2] In particular, same patch size and positional encoding.

Aggregation	Detection									Segmentation								
	Pl. Stool	Rub. Bin	W. Vase	B. Furn.	Cont.	Pl. Vase	W. Stool	Sink Cab.	Mean	Pl. Stool	Rub. Bin	W. Vase	B. Furn.	Cont.	Pl. Vase	W. Stool	Sink Cab.	Mean
max	0.909	0.990	0.945	0.647	0.740	0.607	0.916	1.000	**0.844**	0.855	0.707	0.863	0.791	0.831	0.885	0.711	0.785	**0.804**
min	0.690	1.000	1.000	0.500	0.455	0.606	0.520	1.000	0.721	0.292	0.412	0.847	0.408	0.623	0.874	0.089	0.061	0.451
prod	0.700	1.000	0.972	0.568	0.520	0.618	0.583	1.000	0.745	0.563	0.634	0.881	0.635	0.710	0.869	0.365	0.471	0.641
mean	0.736	1.000	0.972	0.556	0.669	0.613	0.708	1.000	0.782	0.775	0.677	0.878	0.749	0.794	0.865	0.569	0.740	0.756

Table 10. **Aggregation Function.**

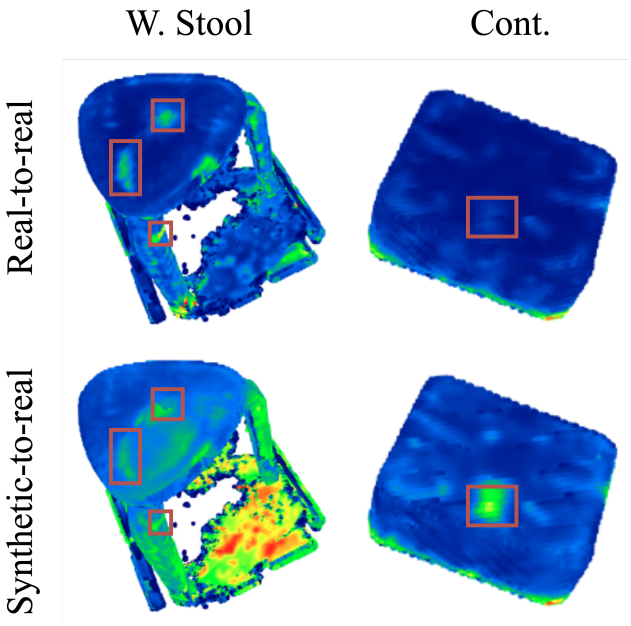

Figure 7. **MODMAP Failure Cases.**

sion gained from our minimum-based cross-view ensembling.

B.3. Visualisations

Failure Cases Fig. 7 illustrates failure cases of MODMAP. In the left example, genuine anomalies (**red boxes**) are overshadowed by spurious peaks arising from unfiltered background regions. This phenomenon is exacerbated in the synthetic-to-real scenario (bottom-left), where the model exhibits particularly pronounced false activations in the unfiltered background due to the domain gap. Indeed, the synthetic training data lacks realistic background appearance, causing the model to misinterpret background variations as anomalies. The right example demonstrates the opposite behaviour: while the real-to-real model (top-right) entirely misses the anomaly, the synthetic-to-real model (bottom-right) correctly identifies and localises the defect. This contrasting behaviour highlights the trade-off between training data fidelity and generalisation. The real-to-real model achieves lower false positive rates by tightly fitting the real normal distribution, but may consequently miss subtle anomalies. The synthetic-to-real model, trained without access to real appearance patterns, maintains higher sensitivity to deviations but struggles to discriminate variations from true defects.

Additional Depth Features Visualisations We report in Fig. 8 the comparison between DINO-v2 and DINO-Depth features, also for the classes missing from the main paper.

Cross-View Maps We report in Fig. 9 and Fig. 10 all the $N \times N$ cross-view visualisations from a test sample of the *Plastic Stool* class from SiM3D.

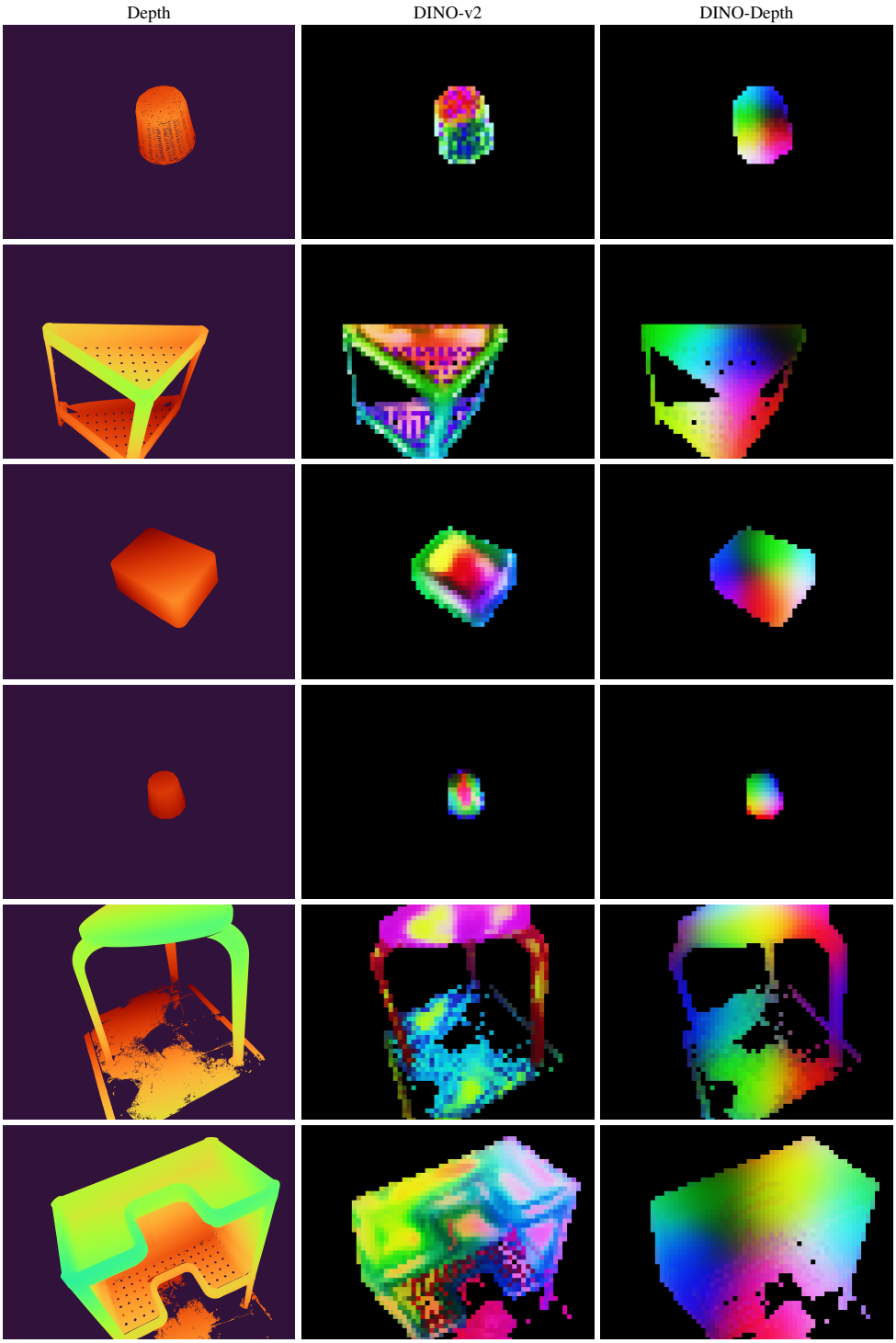

Figure 8. **PCA of Depth Features.**

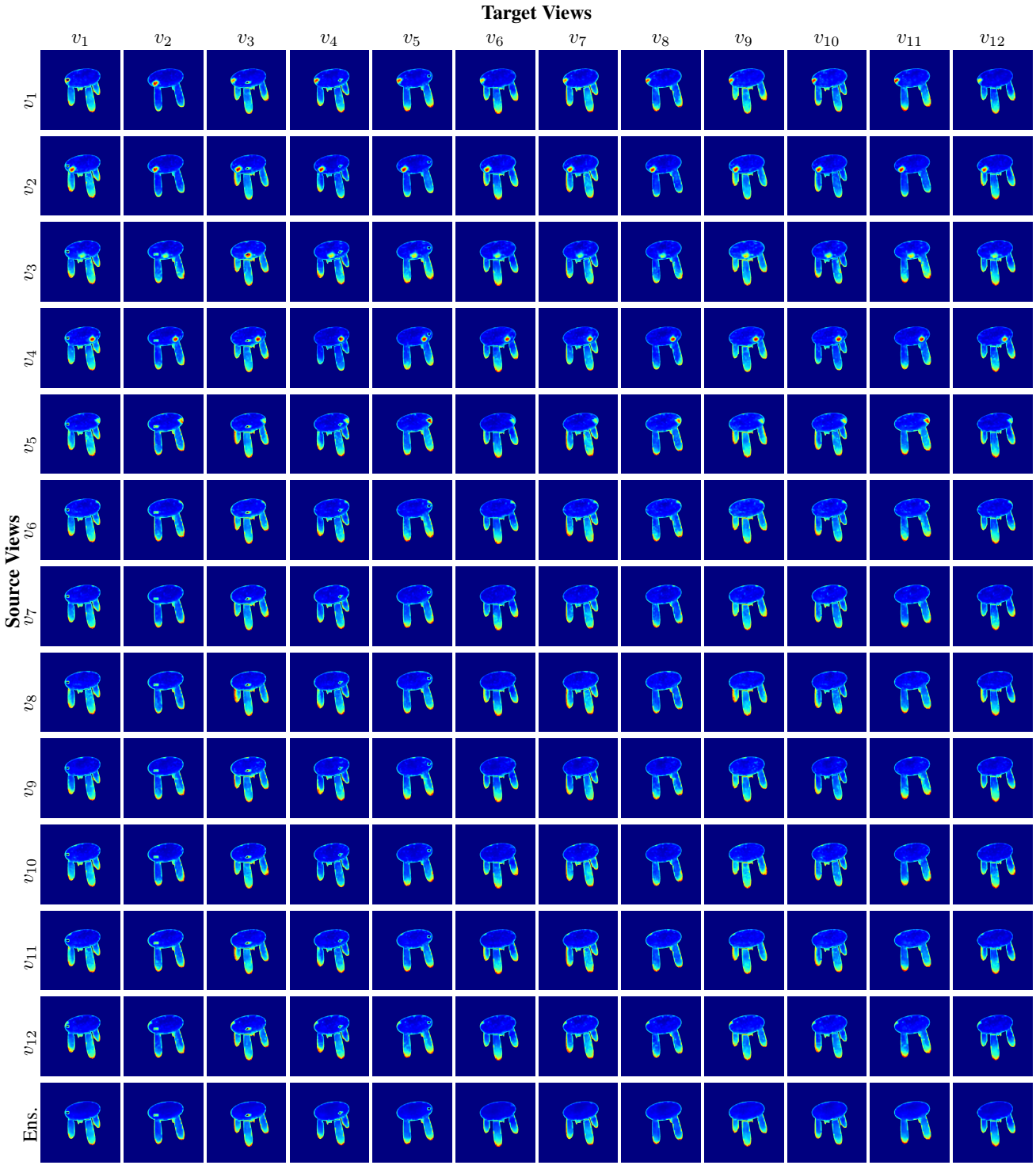

Figure 9. **Image-to-Depth Cross-Views.**

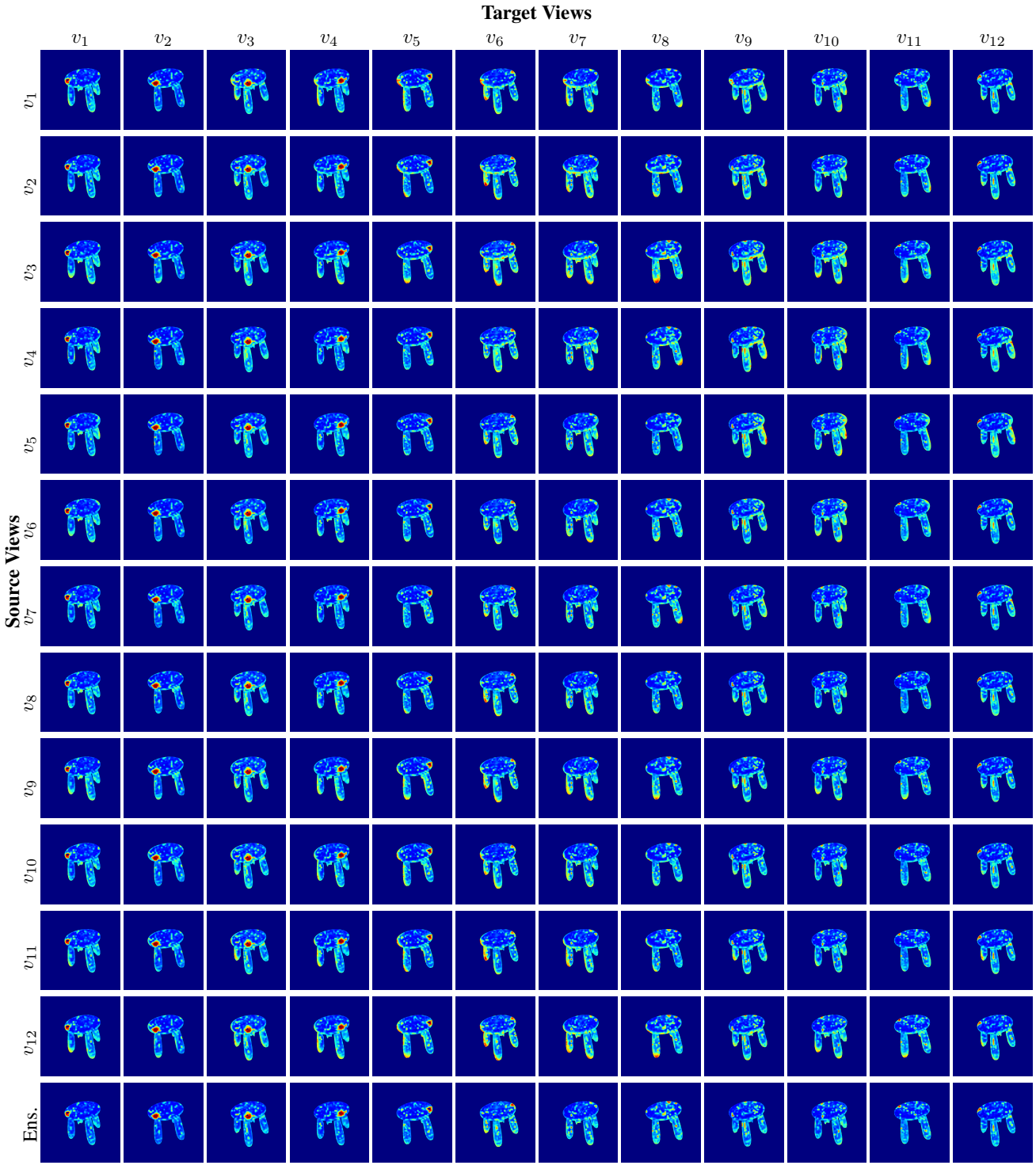

Figure 10. **Depth-to-Image Cross-Views.**