RETHINKING FEED-FORWARD VISUAL PREDICTION WITH META-LEARNED ADAPTATION PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward networks efficiently predict geometry, cameras, and appearance, but can lose accuracy under domain shift. Standard supervised training optimizes prediction without explicitly preparing models to benefit from limited target-domain supervision. We introduce *MetaFeed*, a framework that learns adaptation priors for supervised few-shot domain calibration. Through first-order meta-learning, *MetaFeed* optimizes a shared initialization for prediction after support-set updates to a task-dependent subset of existing weights. Geometry tasks adapt encoder attention projections and feed-forward layers, whereas appearance tasks adapt decoder convolutions. Calibration uses a small labeled support set once per target domain, after which the model processes held-out scenes with fixed weights and ordinary feed-forward inference. Controlled comparisons with supervised training and target fine-tuning under matched data and adaptation protocols show improved cross-domain prediction with competitive in-domain performance. Multi-source controls further show that the calibrated meta-trained model outperforms matched target fine-tuning on a domain unseen during source training. Experiments with VGGT reveal two distinct gains in reconstruction quality: meta-training reduces Chamfer distance before calibration relative to matched supervised training, and supervised calibration using a single support episode from a different scene within the same target domain further reduces it on disjoint target scenes.

1 INTRODUCTION

Deep feed-forward networks predict depth, cameras, point clouds, materials, and segmentation directly from one or multiple images (Wang et al., 2024b; 2025b; Lin et al., 2025; Wu et al., 2025; Li et al., 2025). Their ability to produce predictions in a single forward pass has advanced 3D reconstruction, view synthesis, and embodied perception, with efficient inference that avoids per-scene optimization (Zhang et al., 2025c; Liu et al., 2025; Zhang et al., 2025a).

The performance of these predictors depends on the coverage and quality of their training data (Wang et al., 2026). Collecting supervision for geometry, appearance, and cameras is expensive, and available datasets cover a limited range of environments, sensors, and visual styles. Models can consequently rely on dataset-specific correlations and degrade when deployed under a domain shift (Yu et al., 2024; Tang et al., 2025; Fei et al., 2024; Vuong et al., 2025), as illustrated in Fig. 1.

Increasing source-data diversity improves coverage (Tobin et al., 2017), but cannot be expected to cover every deployment domain. Standard supervision does not explicitly optimize the initialization for few-shot target adaptation, including parameter-efficient fine-tuning (Hu et al., 2021; Zhang et al., 2023). We therefore study whether a dense feed-forward predictor can use a small labeled support set for one-time domain calibration, followed by ordinary feed-forward inference on held-out scenes, and whether the benefits persist under broader source supervision.

We propose *MetaFeed*, a supervised few-shot domain-calibration framework using first-order initialization-based meta-learning (Finn et al., 2017). Support/query episodes optimize a shared initialization for prediction after adaptation. The inner loop updates a task-dependent subset of existing weights: encoder attention projections and feed-forward layers for geometry, and decoder convolutions for appearance. The query loss updates both the shared representation and the adaptable initialization. This design applies to existing feed-forward architectures and retains their parameter count and forward inference procedure after calibration.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

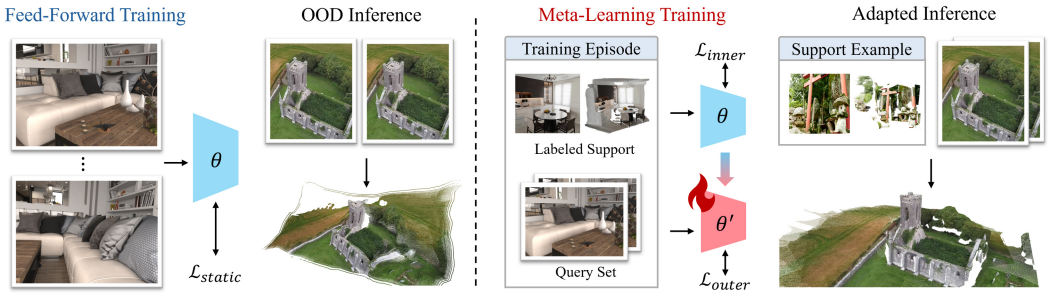


Figure 1: **From static feed-forward prediction to meta-learned fast adaptation.** *Left:* Standard supervised learning converges to a single set of static weights that minimize the average loss over the source domain. This static solution often fails to generalize when the target domain exhibits a large distribution shift. *Right:* Our paradigm learns a transferable initialization θ as an adaptation prior. A small labeled support set adapts only the selected parameter subset in a few inner-loop steps. Calibration is performed once per target domain, followed by fixed-weight prediction on held-out scenes.

Our VGGT controls distinguish two gains in reconstruction quality: meta-training reduces Chamfer distance before calibration relative to matched supervised training, and calibration using a single labeled support episode from a different scene within the same target domain further reduces it on disjoint target scenes.

Our contributions are threefold. First, we formulate dense feed-forward geometry and appearance prediction as supervised few-shot domain calibration. Second, we develop a task-dependent choice of adaptable parameters to address geometric correspondence and appearance decoding within a shared training procedure. Third, controlled comparisons show benefits over standard supervision and target fine-tuning, including on an unseen domain under broader source supervision; VGGT controls separate improvements before calibration from the additional gains of supervised calibration.

2 Related Work

Feed-forward Vision Models. Feed-forward visual prediction replaces iterative geometric optimization with direct neural inference. DUSt3R (Wang et al., 2024b) formulates stereo reconstruction as pointmap regression, while Fast3R (Yang et al., 2025) scales prediction to larger image collections. VGGT (Wang et al., 2025b) jointly predicts cameras, depth, point maps, and tracks from varying numbers of views. WorldMirror (Liu et al., 2025) extends the outputs to surface normals and novel views; MVInverse (Wu et al., 2025) predicts material attributes; and IGGT (Li et al., 2025) addresses multi-view instance segmentation. Their standard supervised objectives do not explicitly optimize these predictors for adaptation from a small target-domain support set.

Meta-Learning and Test-Time Adaptation in Vision. MAML (Finn et al., 2017) learns an initialization for adaptation through a few gradient updates; its first-order approximation omits second-order terms through these updates. ANIL (Raghu et al., 2019) studies feature reuse by restricting inner-loop adaptation to the task head. Domain-oriented approaches, including DAML (Shu et al., 2021), Meta-Causal Learning (Chen et al., 2023), and Arith (Wang et al., 2025d), study related objectives primarily for classification. *MetaFeed* applies first-order initialization-based meta-learning with partial adaptation to dense feed-forward prediction, using task-dependent parameter placement for geometry and appearance. Relative to ordinary fine-tuning or parameter-efficient adaptation such as LoRA (Hu et al., 2021), its shared initialization is explicitly optimized for post-adaptation query performance. Test-time training for 3D reconstruction (Chen et al., 2025; Elflein et al., 2026; Xie et al., 2026; Yuan et al., 2025) adapts models within individual scenes or sequences. *MetaFeed* calibrates once using labeled support from separate scenes within the target domain, then predicts disjoint evaluation scenes with fixed weights.

Few-shot Dense Prediction. VTM (Kim et al., 2023) and Chameleon (Kim et al., 2024) use episodic meta-learning and task-specific adaptation to learn unseen dense prediction tasks, with query predictions conditioned on support image and label tokens. Meta-Initialized-Depth (Wu et al., 2024) learns a depth initialization followed by supervised training on source data for zero-shot cross-dataset inference. *MetaFeed* studies adaptation across domains with the output task fixed: a small labeled support set updates selected existing weights of a geometry or appearance predictor. The adaptation is optimized through support/query episodes, and subsequent queries use the predictor's ordinary forward pass. This setting motivates our comparison of encoder and decoder adaptation for the two output families.

3 METHOD

MetaFeed is a supervised few-shot domain-calibration framework for dense feed-forward prediction. It learns a shared initialization using first-order meta-learning, then adapts a task-dependent subset of existing parameters once per target domain before ordinary feed-forward inference.

3.1 PROBLEM DEFINITION

The input is a set of multi-view images $\{I_v\}_{v=1}^{V}$ captured from a scene, where each $I_v \in \mathbb{R}^{H\times W\times 3}$. The output is a set of dense visual predictions for these views. Depending on the task, the target can be geometric attributes such as point maps, depth maps, and camera parameters, or appearance attributes such as surface normals, albedo, roughness, and metallic maps.

We consider a cross-domain setting where the model is trained on source-domain scenes but evaluated on unseen target-domain scenes. Standard supervised training learns a fixed parameter set θ by minimizing the average loss over source-domain data (Wang et al., 2024b; Yang et al., 2025; Wang et al., 2025b):

$$\min_{\theta} \ \mathbb{E}_{\mathcal{T}\sim p_{\text{src}}(\mathcal{T})}\big[\mathcal{L}(\theta; D_{\mathcal{T}}^{\text{train}})\big], \tag{1}$$

where $\mathcal{T}$ denotes a source-domain task and $D_{\mathcal{T}}^{\text{train}}$ its training views.

During meta-training, we optimize for adaptation through support/query episodes (Finn et al., 2017; Raghu et al., 2019). Each episode $\mathcal{T}_i \sim p(\mathcal{T})$ contains a labeled support set D_i^{sup} and a labeled query set D_i^{qry}, drawn from disjoint views of one scene or disjoint scene subsets within the same source dataset. All episodes optimize one shared initialization $\theta = \{\theta_{\text{shared}}, \phi\}$, where ϕ is a subset of existing network parameters. Episode-specific adaptation is temporary: the support loss updates ϕ, and the query loss evaluates the adapted model. The meta-training objective is:

$$\min_{\theta_{\text{shared}},\,\phi} \ \mathbb{E}_{\mathcal{T}_i\sim p(\mathcal{T})}\big[\mathcal{L}\big(\theta_{\text{shared}}, \phi_i'; D_i^{\text{qry}}\big)\big], \quad \text{where} \quad \phi_i' = \phi - \alpha\nabla_{\phi}\mathcal{L}\big(\theta_{\text{shared}}, \phi; D_i^{\text{sup}}\big). \tag{2}$$

Here α is the inner-loop learning rate. The expression illustrates one adaptation step; the full procedure uses M steps as described in Sec. 3.3. Each episode starts from the same ϕ rather than retaining a separate model for each scene.

3.2 FEED-FORWARD ARCHITECTURE

Backbone. Following VGGT (Wang et al., 2025b), a DINO-based patch embedding (Oquab et al., 2023) maps each image to tokens $\mathbf{X}_v \in \mathbb{R}^{N\times C}$. The Meta Transformer uses L Alternating-Attention blocks to process $\mathbf{X} = \bigcup_{v=1}^{V}\mathbf{X}_v$, alternating frame-wise attention for within-view features and global attention for cross-view correspondence. The Meta DPT Head uses the convolutional DPT decoder (Ranftl et al., 2021) to fuse multi-scale features into dense predictions. Meta-learning changes how these existing weights are trained and adapted.

Task objectives. Geometry outputs are camera intrinsics and extrinsics $\hat{\mathbf{g}}_v$, depth $\hat{\mathbf{D}}_v$, and point maps $\hat{\mathbf{P}}_v$. We use the VGGT objectives (Wang et al., 2025b): a camera Huber loss $\mathcal{L}_{\text{cam}}$ and the DUSt3R uncertainty losses (Wang et al., 2024b) $\mathcal{L}_{\text{depth}}$ and $\mathcal{L}_{\text{pm}}$, with gradient terms for depth and point maps. Appearance outputs are albedo $\hat{\mathbf{A}}_v$, surface normals $\hat{\mathbf{N}}_v$, roughness $\hat{\mathbf{r}}_v$, and metallic $\hat{\mathbf{m}}_v$. Following MVInverse (Wu et al., 2025), we group normals with material attributes, use an angular

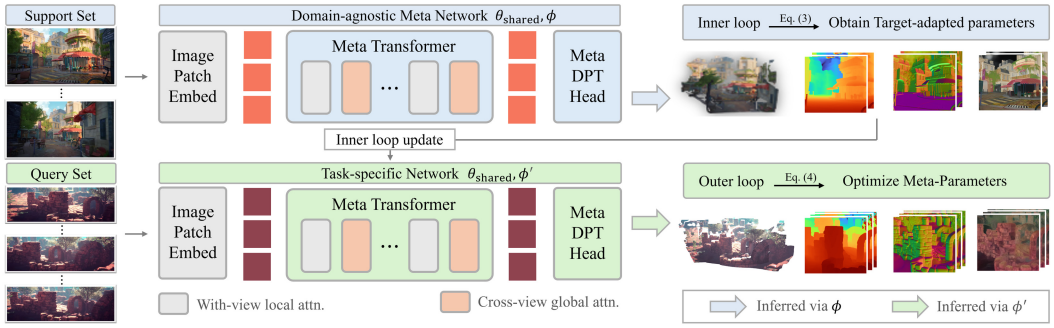


Figure 2: **Overview of *MetaFeed*.** *MetaFeed* provides a meta-learned adaptation prior for fast cross-domain feed-forward visual prediction. Each task contains disjoint support and query sets sampled from the same source dataset. Multi-view images are encoded by the Meta Transformer and decoded by the Meta DPT Head into dense visual predictions. During meta-training, the inner loop adapts a task-dependent subset of existing parameters ϕ using labeled support, keeping θ_{shared} fixed. The outer loop updates both θ_{shared} and the initialization of ϕ using the adapted model's query loss across a meta-batch of tasks.

loss $\mathcal{L}_{\text{normal}}$, and combine MSE and gradient losses for $\mathcal{L}_{\text{alb}}$, $\mathcal{L}_{\text{rough}}$, and $\mathcal{L}_{\text{metal}}$. The active losses form $\mathcal{L}_{\mathcal{T}_i}$; their full definitions are in Appendix A. Both task families use the same meta-learning procedure, with different adaptable parameter subsets.

3.3 Meta-Learning for Fast Domain Adaptation

The inner loop adapts ϕ while keeping θ_{shared} fixed, following the feature-reuse principle of ANIL-style partial adaptation (Raghu et al., 2019). The outer loop learns both parameter groups. Within this procedure, the choice of ϕ determines which parts of prediction can respond to target support. We select existing weights according to the prediction task, preserving the architecture and deployed parameter count.

Task-dependent adaptation prior. For geometry, we place adaptation in the encoder so that target support can update within-view feature processing and cross-view evidence aggregation. We adapt attention projections and feed-forward (FFN) weights: $\phi_{\text{enc}} = \bigcup_{\ell \in \mathcal{E}} \{\mathbf{W}_q^{(\ell)}, \mathbf{W}_k^{(\ell)}, \mathbf{W}_v^{(\ell)}, \mathbf{W}_{\text{ffn}}^{(\ell)}\}$. Here $\mathcal{E}$ denotes selected encoder blocks, and the matrices denote query, key, value, and FFN weights.

For appearance, we instead let target support adjust the mapping from shared multi-view features to material and surface attributes. We adapt DPT decoder convolutions: $\phi_{\text{dec}} = \bigcup_{\ell \in \mathcal{D}} \{\mathbf{W}_{\text{conv}}^{(\ell)}\}$, where $\mathcal{D}$ denotes selected decoder layers, keeping the encoder fixed during calibration. The selected encoder and decoder subsets account for 8.6% and 2.8% of model parameters, respectively. Table 4 supports these task-dependent choices: encoder adaptation is more effective for geometry, while decoder adaptation is more effective for appearance.

Inner-loop adaptation. For a sampled task $\mathcal{T}_i$, we initialize the adaptable parameters as $\phi_i^0 = \phi$ and update them on the support set for M gradient steps:

$$\phi_i^{(m+1)} = \phi_i^{(m)} - \alpha \nabla_{\phi_i^{(m)}} \mathcal{L}_{\mathcal{T}_i}\left(\theta_{\text{shared}}, \phi_i^{(m)}; D_i^{\text{sup}}\right), \quad m = 0, \ldots, M-1, \tag{3}$$

where α is the inner-loop learning rate. The task-adapted parameters for task $\mathcal{T}_i$ are given by $\theta_i' = \{\theta_{\text{shared}}, \phi_i^{(M)}\}$.

Outer-loop meta optimization. For a meta-batch $\mathcal{B} \sim p(\mathcal{T})$, the query objective is $\mathcal{L}_{\text{meta}} = \sum_{\mathcal{T}_i \in \mathcal{B}} \mathcal{L}_{\mathcal{T}_i}(\theta_{\text{shared}}, \phi_i^{(M)}; D_i^{\text{qry}})$. Both the shared parameters and the adaptable initialization are updated:

$$(\theta_{\text{shared}}, \phi) \leftarrow (\theta_{\text{shared}}, \phi) - \beta \nabla_{(\theta_{\text{shared}}, \phi)} \mathcal{L}_{\text{meta}}, \tag{4}$$

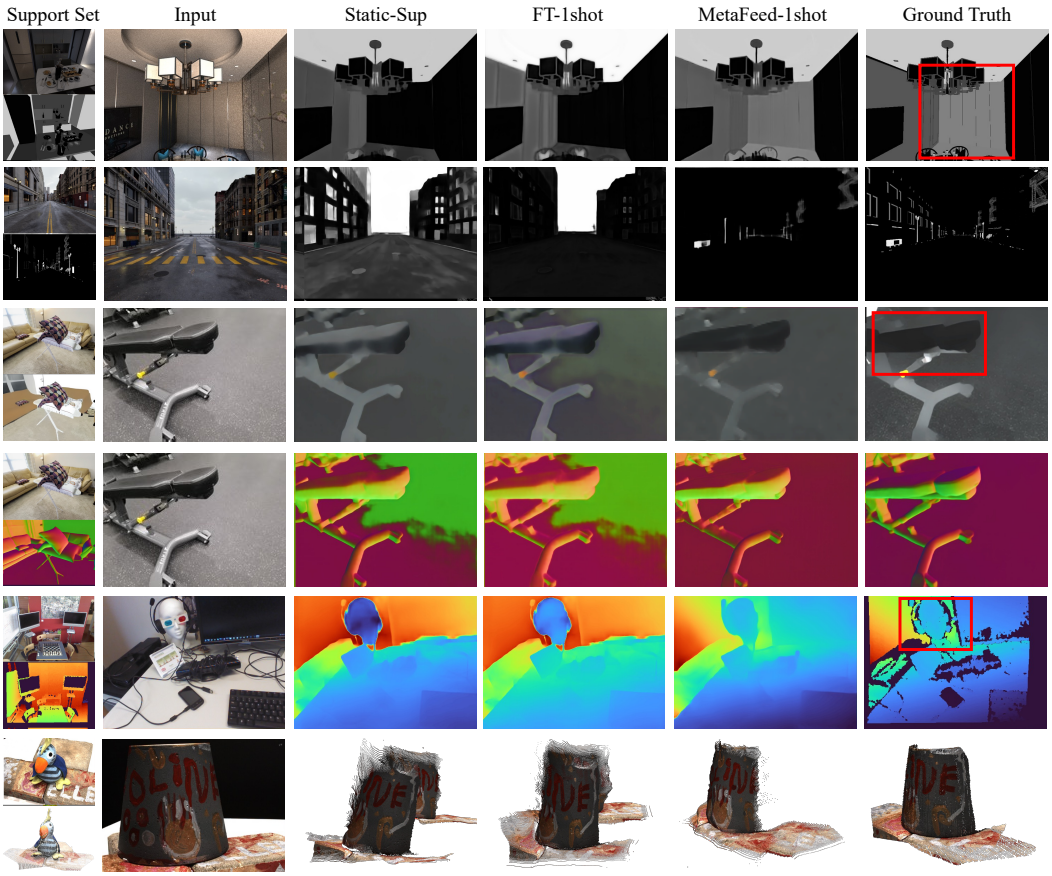


Figure 3: **Qualitative comparison of cross-domain appearance prediction and geometry reconstruction.** The top row shows in-domain InteriorVerse (roughness) predictions. The bottom rows show out-of-domain MatrixCity (metallic), DL3DV (albedo, normal; teacher reference maps), 7Scenes (depth), and DTU (point cloud) predictions. "1shot" denotes one-time calibration using a labeled support episode from a separate scene in the same target domain.

where β is the meta-learning rate. We use the first-order MAML approximation (Finn et al., 2017), omitting second-order terms through the inner-loop updates while retaining query gradients for both parameter groups. Algorithm 1 summarizes the procedure.

3.4 Training and Inference Protocol

At each meta-iteration, we mix intra-scene and cross-scene support/query episodes. Support and query use separate forward passes: the support loss updates only ϕ, and the query pass uses $(\theta_{\text{shared}}, \phi_i^{(M)})$. Query attention receives only query views, so support information is transferred through the parameter update. Episode construction is detailed in Appendix C.1.

At deployment, a support episode consists of labeled views from k support scenes within the same target domain as the evaluation scenes, with support and evaluation scenes disjoint. The default setting uses one support scene ($k = 1$). We calibrate ϕ once per target domain using M inner-loop steps on this support episode, then reuse the fixed adapted weights for ordinary feed-forward prediction on held-out scenes. Support supervision may come from ground truth or teacher predictions; obtaining reliable material supervision for real scenes remains a limitation.

Table 1: **Appearance prediction with independent ground truth.** *FT-k* adapts the Static-Sup decoder parameters and *MetaFeed-k* adapts the same subset from the meta-trained initialization, using the same k support scenes. Best per column within each domain is highlighted in **bold**.

Dataset	Method	Albedo			Roughness	Metallic	Normal	
		PSNR↑	SSIM↑	LPIPS↓	RMSE↓	RMSE↓	Mean↓	Median↓
InteriorVerse *(in-domain)*	*Static-Sup*	21.60	0.862	0.200	**0.123**	0.105	**3.77**	**0.92**
	FT-1	18.12	0.836	0.289	0.201	0.107	14.98	13.19
	FT-3	20.75	0.855	0.245	0.179	0.108	17.59	16.52
	FT-5	18.65	0.842	0.289	0.257	0.116	13.55	10.80
	MetaFeed-1	**21.71**	0.864	0.200	0.125	0.102	4.54	1.86
	MetaFeed-3	21.52	0.863	0.199	0.127	0.103	4.60	1.74
	MetaFeed-5	21.68	**0.864**	**0.199**	0.124	**0.101**	5.13	2.78
MatrixCity *(OOD)*	*Static-Sup*	12.70	0.465	0.536	0.623	0.220	26.76	20.15
	FT-1	14.53	0.460	0.634	0.242	0.154	29.75	23.94
	FT-3	14.63	0.454	0.650	0.227	0.148	28.80	23.03
	FT-5	14.52	0.485	0.621	0.184	0.149	32.72	26.55
	MetaFeed-1	18.27	0.617	0.482	0.173	0.109	26.01	19.83
	MetaFeed-3	18.33	**0.631**	0.514	0.174	**0.105**	**24.02**	17.08
	MetaFeed-5	**18.46**	0.613	**0.472**	**0.169**	0.106	24.12	**16.18**

4 EXPERIMENTS

We evaluate *MetaFeed* through controlled single-source experiments on appearance and geometry (Sec. 4.1), followed by integration into VGGT with matched supervised-training controls (Sec. 4.2). The source-diversity study is presented in Sec. 4.4; additional external-reference and adaptation studies are reported in Appendix E.

4.1 CONTROLLED CROSS-DOMAIN EVALUATION ON A SINGLE DATASET

To isolate the effect of meta-training, we design a controlled setting where the training domain is fixed to one synthetic dataset while test domains span diverse synthetic and real environments. All models share the same backbone (Sec. 3.2) and differ only in the training algorithm: the standard supervised objective (*Static-Sup*) and our meta-optimized scheme, *MetaFeed*. *FT-k* starts from the Static-Sup checkpoint, whereas *MetaFeed-k* starts from the meta-trained initialization. Both adapt the same parameter subset using the same support episode, containing labeled views from k target-domain scenes disjoint from the evaluation scenes, and the same adaptation protocol. The adapted weights remain fixed during evaluation. For geometry, ϕ_{enc} comprises attention projections and feed-forward parameters in the 4th, 11th, 17th, and 23rd encoder blocks. Training and adaptation settings are given in Appendix C.

4.1.1 APPEARANCE PREDICTION

Setup. We train on InteriorVerse (Zhu et al., 2022) using its official split. MatrixCity (Li et al., 2023) provides independent ground truth for out-of-domain appearance evaluation. We evaluate 128 sampled test scenes with four views each per domain. Support episodes use labeled scenes from the same domain, disjoint from the test set. An auxiliary DL3DV (Ling et al., 2024) teacher-agreement study is reported in Appendix D.

Results. Table 1 reports albedo PSNR/SSIM/LPIPS, roughness and metallic RMSE, and normal mean/median angular error. *MetaFeed* retains competitive in-domain performance and improves MatrixCity predictions over matched fine-tuning, including with one support scene. Source-data diversity is studied in Sec. 4.4. External references and ordinary distillation are compared in Appendix E.1; teacher-supported calibration and its accuracy–speed trade-off are studied in Appendix E.3. Figures 3 and 4 provide qualitative examples.

Table 2: **Cross-domain geometry evaluation.** *Static-Sup* is the standard supervised baseline. *FT-k* and *MetaFeed-k* adapt the same encoder attention/FFN parameter subset from supervised and meta-trained initializations, respectively, using the same k support scenes. Best per column within each domain is highlighted in **bold**.

Dataset	Method	Point Cloud			Depth		Camera Pose		
		Acc↓	Comp↓	CD↓	AbsRel↓	δ<1.25↑	AUC↑	RRA↑	RTA↑
HyperSim *(in-domain)*	*Static-Sup*	0.038	0.051	0.045	0.078	0.922	0.694	0.948	0.896
	FT-1	0.039	0.052	0.046	0.087	0.920	0.676	0.953	0.891
	FT-3	0.039	0.052	0.045	0.082	0.923	0.683	0.953	0.896
	FT-5	0.039	0.052	0.046	0.083	0.923	0.681	0.948	0.891
	MetaFeed-1	**0.035**	0.048	**0.041**	**0.078**	**0.930**	**0.712**	**0.958**	**0.906**
	MetaFeed-3	0.035	**0.047**	**0.041**	**0.078**	0.925	0.705	0.953	0.901
	MetaFeed-5	0.038	0.051	0.044	0.093	0.909	0.698	0.953	0.896
DTU *(OOD)*	*Static-Sup*	8.459	34.141	21.300	0.086	0.918	0.026	0.363	0.208
	FT-1	9.774	25.757	17.765	0.051	0.979	0.029	0.316	0.245
	FT-3	9.818	24.253	17.035	0.050	0.980	0.028	0.309	0.244
	FT-5	9.850	24.085	16.968	0.050	0.980	0.029	0.311	0.246
	MetaFeed-1	**7.737**	12.287	10.012	0.039	0.990	**0.038**	**0.392**	**0.261**
	MetaFeed-3	8.312	**8.200**	**8.256**	**0.039**	**0.991**	0.035	0.385	0.253
	MetaFeed-5	8.457	9.183	8.820	0.041	**0.991**	0.032	0.378	0.248
7Scenes *(OOD)*	*Static-Sup*	0.130	0.467	0.298	0.280	0.842	0.104	0.448	0.359
	FT-1	0.141	0.456	0.298	0.271	0.838	0.080	0.444	0.348
	FT-3	0.132	0.461	0.296	0.271	0.852	0.099	0.456	0.359
	FT-5	0.139	0.469	0.304	0.270	0.848	0.093	0.448	0.359
	MetaFeed-1	0.113	**0.210**	0.161	0.248	0.902	**0.118**	**0.472**	**0.378**
	MetaFeed-3	**0.105**	0.213	**0.159**	0.245	**0.913**	0.113	0.463	0.371
	MetaFeed-5	0.105	0.230	0.168	**0.243**	**0.913**	0.110	0.460	0.367

4.1.2 Geometry Reconstruction

Setup. We adopt HyperSim (Roberts et al., 2021) as the synthetic indoor source domain, which provides dense depth, camera poses, and point cloud ground truth. For out-of-domain evaluation, we select the real-world object-centric multi-view dataset DTU (Jensen et al., 2014) and the real indoor RGB-D 7Scenes (Shotton et al., 2013), both exhibiting significant domain gaps from HyperSim in camera configurations, scene scale, and imaging conditions. We follow the multi-view evaluation protocol of Wang et al. (2025b). The support episode uses separate scenes from the same target domain as the test scenes.

Results. Table 2 reports cross-domain results on point cloud reconstruction, depth estimation, and camera pose estimation. We report Accuracy, Completeness, and CD (Chamfer Distance) for point clouds; AbsRel and $\delta < 1.25$ for depth; and RRA, RTA, and AUC@30° for camera pose. *MetaFeed* retains competitive in-domain performance and improves the evaluated geometry tasks on DTU and 7Scenes. Compared with fine-tuning the same parameter subset on the same support scenes, its stronger post-adaptation performance supports the benefit of the meta-trained initialization. Figures 3 and 4 illustrate improvements in reconstructed geometry.

4.2 Boosting State-of-the-Art Feed-Forward Models

We apply *MetaFeed* to VGGT (Wang et al., 2025b) to test whether meta-training improves an existing feed-forward geometry model after training on a broader synthetic corpus.

Setup. We retain the VGGT architecture and adapt its selected encoder attention/FFN parameters. Fast3R, CUT3R, FLARE, and VGGT use their public checkpoints. VGGT-FT uses standard supervision on the same nine synthetic datasets as *MetaFeed*. VGGT-FT-1 and *MetaFeed*-1 use the same parameter subset and adaptation protocol with a single support episode from one labeled scene in the target domain, disjoint from evaluation scenes. We evaluate on 7-Scenes (Shotton et al., 2013),

Table 3: **Geometry transfer and matched training controls.** Lower is better for Accuracy (Acc.), Completeness (Comp.), and Chamfer Distance (CD). Public checkpoints appear above the divider; VGGT-FT and *MetaFeed* use supervised and meta-training, respectively, on the same nine synthetic datasets. The suffix "1" denotes calibration using a labeled one-scene target support episode, disjoint from evaluation scenes; other rows are uncalibrated. Best values are in **bold**.

Method	7-Scenes			NRGBD			DTU		
	Acc.↓	Comp.↓	CD↓	Acc.↓	Comp.↓	CD↓	Acc.↓	Comp.↓	CD↓
Fast3R (Yang et al., 2025)	0.096	0.145	0.120	0.135	0.163	0.149	3.340	2.929	3.135
CUT3R (Wang et al., 2025c)	0.094	0.101	0.098	0.104	0.079	0.091	4.742	3.400	4.071
FLARE (Zhang et al., 2025b)	0.085	0.142	0.113	0.053	0.051	0.052	2.541	3.174	2.857
VGGT (Wang et al., 2025b)	0.046	0.057	0.052	0.051	0.066	0.058	1.338	1.896	1.617
VGGT-FT	0.052	0.070	0.061	0.053	0.044	0.048	1.652	2.132	1.892
VGGT-FT-1	0.050	0.069	0.059	0.053	0.045	0.049	1.673	2.079	1.876
MetaFeed	0.044	0.052	0.048	0.052	**0.041**	0.047	1.417	1.254	1.335
MetaFeed-1	**0.038**	**0.042**	**0.040**	**0.042**	0.044	**0.043**	**1.116**	**0.921**	**1.018**

Table 4: **Task-dependent placement of adaptable parameters.** Encoder adaptation is more effective for geometry, while decoder adaptation is more effective for appearance on MatrixCity. Best values per column are in **bold**.

Method	Geometry				Appearance		
	Point Cloud		Depth		Albedo	Roughness	Metallic
	Acc↓	Comp↓	AbsRel↓	δ<1.25↑	PSNR↑	RMSE↓	RMSE↓
Static-Sup	0.130	0.467	0.280	0.842	12.70	0.623	0.220
MetaFeed w/ ϕ_{enc}	**0.113**	**0.210**	**0.248**	**0.902**	16.85	0.196	0.134
MetaFeed w/ ϕ_{dec}	0.121	0.474	0.271	0.827	**18.27**	**0.173**	**0.109**

NRGBD (Azinović et al., 2022), and DTU (Jensen et al., 2014). The training setup is described in Appendix C.

Results. Table 3 distinguishes two gains on 7-Scenes/NRGBD/DTU. Relative to same-data VGGT-FT, meta-training reduces CD from 0.061/0.048/1.892 to 0.048/0.047/1.335 before calibration. Calibration using a single supervised support episode from a separate scene in the same target domain further reduces CD to 0.040/0.043/1.018 on disjoint test scenes. LoRA uses less adaptation time and memory, while *MetaFeed* achieves lower CD under the same support/evaluation protocol (Appendix E.2). Figure 5 visualizes calibrated reconstruction.

4.3 Ablations

Adaptation settings. We select hyperparameters in preliminary validation and fix $(\alpha, K, M) = (0.1, 1, 5)$ for controlled experiments and $(0.05, 1, 5)$ for VGGT across target domains. Five-seed support-sampling results and support-size, step-count, and learning-rate sweeps are reported in Appendix E.4. Training cost and one-time calibration overhead are detailed in Appendix G.

Task-dependent targeted adaptation. Table 4 supports the parameter choices in Sec. 3.3. Encoder adaptation lowers geometry depth AbsRel to 0.248 versus 0.271 for decoder adaptation. On MatrixCity, decoder adaptation raises albedo PSNR from 16.85 to 18.27 and lowers roughness/metallic RMSE from 0.196/0.134 to 0.173/0.109 relative to encoder adaptation, while updating fewer parameters.

4.4 Source-Data Diversity

We test adaptation under broader source supervision. Single-source training uses InteriorVerse; multi-source training adds MatrixCity at a 1:1 ratio. MatrixCity is then source-covered, while Structured3D (Zheng et al., 2020) remains unseen.

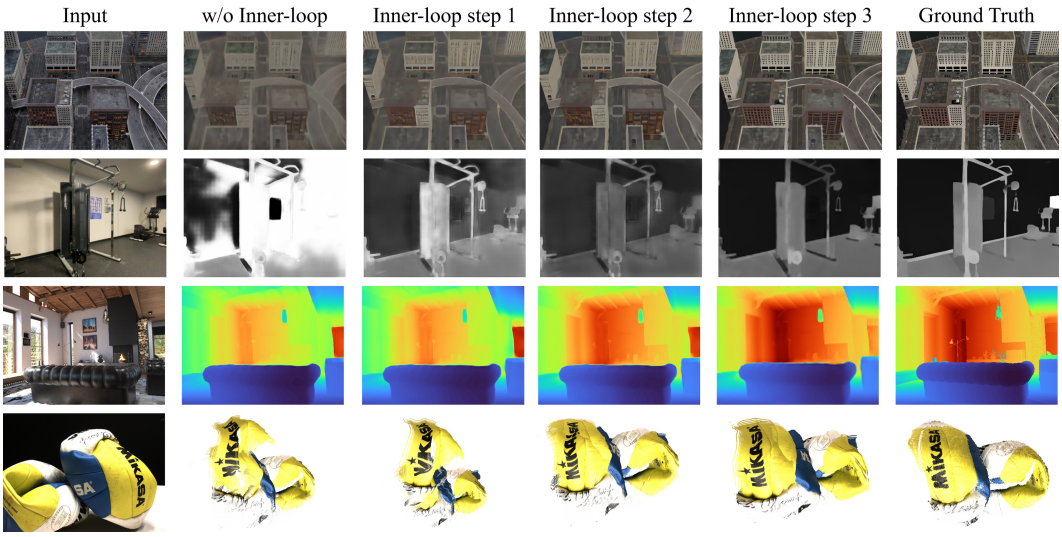


Figure 4: **Inner-loop adaptation** of albedo, metallic, depth, and point cloud predictions under the controlled experimental setting in Sec. 4.1. Predictions improve within a few supervised inner-loop updates, illustrating rapid adaptation to new target domains.

Table 5: **Single- and multi-source appearance training.** MatrixCity is source-covered in multi-source training; Structured3D remains unseen in both regimes. Methods within each regime use identical source data. FT-1 and *MetaFeed*-1 share the same support episode from one labeled target-domain scene and the same adaptation/evaluation protocol; support and test scenes are disjoint.

Training	**Method**	**MatrixCity**			**Structured3D**		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Single-source	Static-Sup	12.70	0.465	0.536	14.63	0.620	0.358
	FT-1	14.53	0.460	0.634	15.41	0.652	0.296
	MetaFeed-1	**18.27**	**0.617**	**0.482**	**17.86**	**0.756**	**0.231**
Multi-source	Static-Sup	19.24	0.712	**0.260**	15.85	0.648	0.274
	FT-1	18.46	0.691	0.279	16.28	0.662	0.252
	MetaFeed-1	**19.31**	**0.720**	0.263	**19.52**	**0.808**	**0.186**

In Table 5, multi-source Static-Sup nearly matches *MetaFeed*-1 on MatrixCity. On unseen Structured3D, *MetaFeed*-1 achieves 19.52 PSNR versus 16.28 for FT-1 with matched source data, decoder parameters, target support, and adaptation protocols. The calibrated meta-trained model thus retains an advantage over matched fine-tuning under broader source supervision.

5 Conclusion

We introduced *MetaFeed*, a supervised few-shot domain-calibration framework for dense feed-forward geometry and appearance prediction. First-order meta-learning optimizes a shared initialization, with encoder attention and feed-forward layers adapted for geometry and decoder convolutions adapted for appearance. Controlled comparisons show benefits over standard supervision and matched target fine-tuning, including on a target domain unseen during multi-source training. The VGGT controls distinguish two gains: meta-training lowers Chamfer distance before calibration relative to matched supervised training, and supervised calibration using a single support episode from a different scene within the same target domain further reduces it. After this one-time calibration, the model predicts disjoint target scenes through ordinary feed-forward inference with fixed weights.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks requiring disclosure under the ICLR 2027 AI Policy for Authors. We used generative AI tools for language editing and for creating or editing software code. The authors reviewed all AI-assisted work and take responsibility for the final manuscript, code, and claims. We have reviewed all AI-assisted work. All AI-assisted edits were carefully reviewed by the authors to ensure correctness and to preserve the intended technical meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6290–6301, 2022.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Jin Chen, Zhi Gao, Xinxiao Wu, and Jiebo Luo. Meta-causal learning for single domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7683–7692, 2023.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.

Sven Elflein, Ruilong Li, Sérgio Agostinho, Zan Gojcic, Laura Leal-Taixé, Qunjie Zhou, and Aljosa Osep. Vgg-t^3: Offline feed-forward 3d reconstruction at scale. *arXiv preprint arXiv:2602.23361*, 2026.

Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Driv3r: Learning dense 4d reconstruction for autonomous driving. *arXiv preprint arXiv:2412.06777*, 2024.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Ning Gao, Hanna Ziesche, Ngo Anh Vien, Michael Volpp, and Gerhard Neumann. What matters for meta-learning vision regression tasks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14776–14786, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 406–413, 2014.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

Donggyun Kim, Jinwoo Kim, Seongwoong Cho, Chong Luo, and Seunghoon Hong. Universal few-shot learning of dense prediction tasks with visual token matching. *arXiv preprint arXiv:2303.14969*, 2023.

Donggyun Kim, Seongwoong Cho, Semin Kim, Chong Luo, and Seunghoon Hong. Chameleon: A data-efficient generalist for dense visual prediction in the wild. In *European Conference on Computer Vision*, pp. 422–441. Springer, 2024.

Hao Li, Zhengyu Zou, Fangfu Liu, Xuanyang Zhang, Fangzhou Hong, Yukang Cao, Yushi Lan, Manyuan Zhang, Gang Yu, Dingwen Zhang, et al. Iggt: Instance-grounded geometry transformer for semantic 3d reconstruction. *arXiv preprint arXiv:2510.22706*, 2025.

Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3205–3215, 2023.

Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Chih-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26069–26080. IEEE, 2025.

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.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Huan Liu, Zhixiang Chi, Yuanhao Yu, Yang Wang, Jun Chen, and Jin Tang. Meta-auxiliary learning for future depth prediction in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5756–5765, 2023.

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20133–20143, 2023.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12179–12188, 2021.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10912–10922, 2021.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9339–9347, 2019.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2930–2937, 2013.

Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9624–9633, 2021.

Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5283–5293, 2025.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8942–8952, 2021.

Riccardo Volpi, Diane Larlus, and Grégory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4443–4453, 2021.

Khiem Vuong, Anurag Ghosh, Deva Ramanan, Srinivasa Narasimhan, and Shubham Tulsiani. Aerialmegadepth: Learning aerial-ground reconstruction and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21674–21684, 2025.

Huaqing Wang, Dongrui Lv, Tianjiao Lin, Changkun Han, and Liuyang Song. Task-adaptive unbiased regularization meta-learning for few-shot cross-domain fault diagnosis. *Engineering Applications of Artificial Intelligence*, 144:110200, 2025a.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21686–21697, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025b.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10510–10522, 2025c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024b.

Weijie Wang, Qihang Cao, Sensen Gao, Donny Y Chen, Haofei Xu, Wenjing Bian, Songyou Peng, Tat-Jen Cham, Chuanxia Zheng, Andreas Geiger, et al. Feed-forward 3d scene modeling: A problem-driven perspective. *arXiv preprint arXiv:2604.14025*, 2026.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916. IEEE, 2020.

Xiran Wang, Jian Zhang, Lei Qi, and Yinghuan Shi. Balanced direction from multifarious choices: Arithmetic meta-learning for domain generalization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30577–30587, 2025d.

Cho-Ying Wu, Yiqi Zhong, Junying Wang, and Ulrich Neumann. Boosting generalizability towards zero-shot cross-dataset single-image indoor depth by meta-initialization. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10051–10058. IEEE, 2024.

Xiangzuo Wu, Chengwei Ren, Jun Zhou, Xiu Li, and Yuan Liu. Mvinverse: Feed-forward multi-view inverse rendering in seconds. *arXiv preprint arXiv:2512.21003*, 2025.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, et al. Scal3r: Scalable test-time training for large-scale 3d reconstruction. *arXiv preprint arXiv:2604.08542*, 2026.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21924–21935, 2025.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1790–1799, 2020.

Han Yu, Jiashuo Liu, Xingxuan Zhang, Jiayun Wu, and Peng Cui. A survey on evaluation of out-of-distribution generalization. *arXiv preprint arXiv:2403.01874*, 2024.

Yuheng Yuan, Qiuhong Shen, Shizun Wang, Xingyi Yang, and Xinchao Wang. Test3r: Learning to reconstruct 3d at test time. *arXiv preprint arXiv:2506.13750*, 5, 2025.

Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. Rgb↔ x: Image decomposition and synthesis using material-and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K Barral, Raia Hadsell, et al. Efficiently reconstructing dynamic scenes one d4rt at a time. *arXiv preprint arXiv:2512.08924*, 2025a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21936–21947, 2025b.

Wei Zhang, Yihang Wu, Songhua Li, Wenjie Ma, Xin Ma, Qiang Li, and Qi Wang. Review of feed-forward 3d reconstruction: From dust3r to vggt. *arXiv preprint arXiv:2507.08448*, 2025c.

Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision*, pp. 519–535. Springer, 2020.

Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19855–19865, 2023.

Jingsen Zhu, Fujun Luan, Yuchi Huo, Zihao Lin, Zhihua Zhong, Dianbing Xi, Rui Wang, Hujun Bao, Jiaxiang Zheng, and Rui Tang. Learning-based inverse rendering of complex indoor scenes with differentiable monte carlo raytracing. In *Siggraph asia 2022 conference papers*, pp. 1–8, 2022.

A Loss Function Definitions

This appendix provides the full mathematical definitions of the loss terms used in Sec. 3.2. We use V to denote the number of input views, and Ω_v to denote the set of valid pixels in the v-th frame.

A.1 Geometry Losses

The total geometry loss is the unweighted sum of the camera loss, depth loss, and point map loss:

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pm}}. \tag{5}$$

Camera loss. Following VGGT (Wang et al., 2025b), we supervise the predicted camera parameters $\hat{\mathbf{g}}_v$ against the ground-truth $\mathbf{g}_v^*$ with a Huber loss:

$$\mathcal{L}_{\text{cam}} = \sum_{v=1}^{V} \left\| \hat{\mathbf{g}}_v - \mathbf{g}_v^* \right\|_\epsilon, \tag{6}$$

where $\|\cdot\|_\epsilon$ denotes the Huber norm with threshold ϵ. Each $\mathbf{g}_v$ contains the camera rotation, translation, and field of view, parametrized as in VGGSfM (Wang et al., 2024a).

Depth loss. Following DUSt3R (Wang et al., 2024b) and VGGT (Wang et al., 2025b), depth predictions are supervised by an aleatoric-uncertainty loss with an additional gradient term:

$$\mathcal{L}_{\text{depth}} = \sum_{v=1}^{V} \left\| \boldsymbol{\Sigma}_v^D \odot (\hat{\mathbf{D}}_v - \mathbf{D}_v^*) \right\| + \left\| \boldsymbol{\Sigma}_v^D \odot (\nabla \hat{\mathbf{D}}_v - \nabla \mathbf{D}_v^*) \right\| - \lambda_{\text{unc}} \log \boldsymbol{\Sigma}_v^D, \tag{7}$$

where $\boldsymbol{\Sigma}_v^D \in \mathbb{R}_+^{H \times W}$ is the predicted aleatoric uncertainty map, ∇ denotes the spatial gradient, $\odot$ is the channel-broadcast element-wise product, and λ_{unc} is a scalar uncertainty-loss coefficient, distinct from the inner-loop learning rate α.

Point map loss. The point map loss is defined analogously to the depth loss, with an independently predicted uncertainty map $\boldsymbol{\Sigma}_v^P \in \mathbb{R}_+^{H \times W}$:

$$\mathcal{L}_{\text{pm}} = \sum_{v=1}^{V} \left\| \boldsymbol{\Sigma}_v^P \odot (\hat{\mathbf{P}}_v - \mathbf{P}_v^*) \right\| + \left\| \boldsymbol{\Sigma}_v^P \odot (\nabla \hat{\mathbf{P}}_v - \nabla \mathbf{P}_v^*) \right\| - \lambda_{\text{unc}} \log \boldsymbol{\Sigma}_v^P. \tag{8}$$

A.2 Appearance Losses

The total appearance loss is the unweighted sum of the surface normal loss and the per-modality material losses:

$$\mathcal{L}_{\text{app}} = \mathcal{L}_{\text{normal}} + \mathcal{L}_{\text{alb}} + \mathcal{L}_{\text{rough}} + \mathcal{L}_{\text{metal}}. \tag{9}$$

Surface normal loss. Surface normals are supervised with an Angular Loss (AL) that penalizes the cosine deviation between predicted and ground-truth unit normals:

$$\mathcal{L}_{\text{normal}} = \sum_{v=1}^{V} \frac{1}{|\Omega_v|} \sum_{p \in \Omega_v} \left(1 - \hat{\mathbf{N}}_v(p) \cdot \mathbf{N}_v^*(p)\right). \tag{10}$$

Albedo, roughness, and metallic losses. Albedo, roughness, and metallic are each supervised with a combined MSE and gradient loss. Letting $\mathbf{Y} \in \{\mathbf{A}, \mathbf{r}, \mathbf{m}\}$ denote a generic appearance target, the loss is

$$\mathcal{L}_{\text{Y}} = \sum_{v=1}^{V} \left\| \hat{\mathbf{Y}}_v - \mathbf{Y}_v^* \right\|_2^2 + \left\| \nabla \hat{\mathbf{Y}}_v - \nabla \mathbf{Y}_v^* \right\|_2^2, \tag{11}$$

Eq. (11) is instantiated for $\mathbf{Y} = \mathbf{A}$, $\mathbf{Y} = \mathbf{r}$, and $\mathbf{Y} = \mathbf{m}$ to obtain $\mathcal{L}_{\text{alb}}$, $\mathcal{L}_{\text{rough}}$, and $\mathcal{L}_{\text{metal}}$, respectively.

Algorithm 1: First-order meta-training for *MetaFeed*

Input: Multi-view training datasets $\{\mathcal{D}_k\}$; inner-loop step choices $\{3, 5, 7, 10\}$; learning rates α and β.

Output: Meta-trained parameters $\theta = \{\theta_{\text{shared}}, \phi\}$.

Select ϕ according to the target prediction task;
while *not converged* **do**
  Sample a meta-batch of B episodes $\{\mathcal{T}_i\}_{i=1}^{B}$ using intra-scene or cross-scene splits;
  Construct disjoint support/query sets $\{(D_i^{\text{sup}}, D_i^{\text{qry}})\}_{i=1}^{B}$;
  Sample $M \in \{3, 5, 7, 10\}$;
  foreach *episode* $\mathcal{T}_i$ **do**
    Set $\phi_i^0 \leftarrow \phi$;
    for $m = 0$ **to** $M - 1$ **do**
      Forward only D_i^{sup} with $(\theta_{\text{shared}}, \phi_i^m)$ and update ϕ_i^{m+1} using Eq. (3);
    Forward only D_i^{qry} with $(\theta_{\text{shared}}, \phi_i^M)$ and compute $\mathcal{L}_{\mathcal{T}_i}\left(\theta_{\text{shared}}, \phi_i^M; D_i^{\text{qry}}\right)$;
  Use first-order meta-gradients for both outer updates;
  $\mathcal{L}_{\text{meta}} \leftarrow \sum_{i=1}^{B} \mathcal{L}_{\mathcal{T}_i}\left(\theta_{\text{shared}}, \phi_i^M; D_i^{\text{qry}}\right)$;
  $\theta_{\text{shared}} \leftarrow \theta_{\text{shared}} - \beta \nabla_{\theta_{\text{shared}}} \mathcal{L}_{\text{meta}}$;
  $\phi \leftarrow \phi - \beta \nabla_{\phi} \mathcal{L}_{\text{meta}}$;
return $\theta = \{\theta_{shared}, \phi\}$

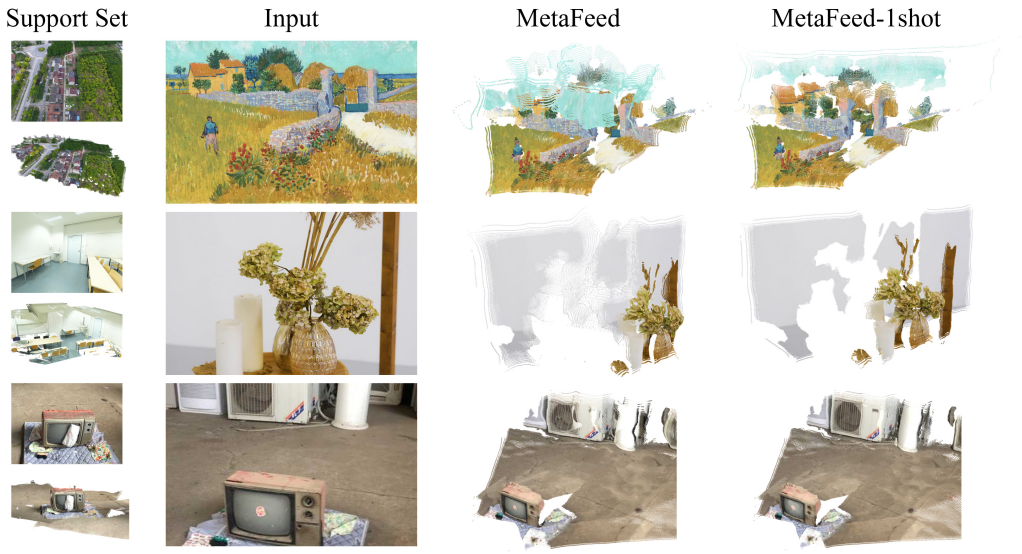


Figure 5: **Visualization on in-the-wild data.** Qualitative comparison between *MetaFeed* without inner-loop adaptation and one support set adaptation on out-of-distribution scenes. The adapted model effectively suppresses geometric noise and resolves finer scene details, demonstrating the benefit of the meta-learned adaptation prior for robust cross-domain generalization.

B META-TRAINING ALGORITHM

We present the complete meta-training loop in Algorithm 1. Each episode uses an intra-scene or cross-scene support/query split (Appendix C). The inner loop adapts existing parameters ϕ on the support set via M gradient steps; the query loss then updates both θ_{shared} and the shared initialization ϕ. Support and query are separate forward passes: query attention contains only query views. First-order meta-gradients omit derivatives through the support gradients while retaining the direct dependence on the initialization. Adapted episode weights are temporary; training does not create a persistent model for each scene.

C Training and Evaluation Settings

C.1 Implementation Details

Architecture. For controlled experiments in Sec. 4.1, dense map outputs for all modalities are predicted by a single unified DPT head, following the design of MapAnything (Keetha et al., 2025). The dense maps for appearance tasks include albedo, roughness, metallic, and normal maps, while those for geometry tasks comprise point maps, depth maps, and confidence maps. For geometry tasks, we adopt a camera token and a dedicated camera head to estimate camera poses and intrinsics (Wang et al., 2025b). The patch embedding layer and the Alternating-Attention transformer follow VGGT (Wang et al., 2025b). We initialize only the patch embedding from DINOv2 (Oquab et al., 2023) weights, following VGGT, while the transformer and DPT head are trained from scratch.

For the boosting SOTA experiments in Sec. 4.2, we apply *MetaFeed* to VGGT (Wang et al., 2025b). We keep the original backbone architecture unchanged and only replace the standard supervised training objective with our meta-learning objective. The selected encoder attention/FFN parameters ϕ_{enc} are selected as the meta-adaptable parameter subset.

Training data. The controlled appearance and geometry experiments use InteriorVerse and Hypersim, respectively, as their single source dataset. Static-Sup and *MetaFeed* use identical source data within each setting. The separate single-/multi-source appearance comparison in Sec. 4.4 also matches training data across methods within each regime. MatrixCity is source-covered in the multi-source regime, whereas Structured3D (Zheng et al., 2020) remains unseen.

For the state-of-the-art model boosting experiments, we construct a large-scale and diverse synthetic training corpus from nine datasets: BlenderMVS (Yao et al., 2020), TartanAir (Wang et al., 2020), ASE (Pan et al., 2023), Hypersim (Roberts et al., 2021), Unreal4K (Tosi et al., 2021), Habitat (Savva et al., 2019), MVS-Synth (Huang et al., 2018), PointOdyssey (Zheng et al., 2023), and VirtualKITTI (Cabon et al., 2020). These datasets cover a broad range of synthetic domains, including indoor scenes, outdoor environments, aerial views, and driving scenarios. We deliberately exclude all real-world data during this training stage. VGGT-FT uses standard supervision on this same nine-dataset corpus, whereas *MetaFeed* uses the meta-learning objective. VGGT-FT-1 further adapts the supervised model using the same target support and evaluation protocol as *MetaFeed*-1. The other feed-forward models in Table 3 are evaluated using their public checkpoints.

Meta-training episode construction. Each meta-training episode consists of a support set D_i^{sup} and a query set D_i^{qry}. We construct these sets using two complementary strategies.

First, in the *intra-scene split*, support and query samples are drawn from different views of the same scene. Formally, for a scene $\mathcal{S}_i$ with a set of posed views $\mathcal{V}_i = \{I_{i,v}\}_{v=1}^{V_i}$, we partition the views into two disjoint subsets,

$$D_i^{\text{sup}} = \{I_{i,v} \mid v \in \mathcal{A}_i\}, \qquad D_i^{\text{qry}} = \{I_{i,v} \mid v \in \mathcal{B}_i\}, \qquad \mathcal{A}_i \cap \mathcal{B}_i = \emptyset. \tag{12}$$

This split encourages the model to adapt to a specific 3D environment from a small number of support views and to generalize to unseen views of the same scene.

Second, in the *cross-scene split*, support and query samples are drawn from different scenes within the same dataset. Given a dataset $\mathcal{D}_k$ containing scenes $\{\mathcal{S}_{k,n}\}_{n=1}^{N_k}$, we sample two disjoint scene subsets $\mathcal{P}_i$ and $\mathcal{Q}_i$ and define

$$D_i^{\text{sup}} = \{I_{k,n,v} \mid \mathcal{S}_{k,n} \in \mathcal{P}_i\}, \qquad D_i^{\text{qry}} = \{I_{k,n,v} \mid \mathcal{S}_{k,n} \in \mathcal{Q}_i\}, \qquad \mathcal{P}_i \cap \mathcal{Q}_i = \emptyset. \tag{13}$$

This split introduces a stronger distribution shift between support and query data, encouraging the learned initialization to capture more domain-level geometric priors rather than overfitting to a single scene. During training, we mix the two splitting strategies to jointly improve view-level adaptation and scene-level generalization. All episodes start from one shared initialization. Support views are processed with the current inner-loop weights, and query views are processed separately with the adapted weights; support views are never included as query attention context.

Optimization. For controlled experiments, models are trained for 20k steps on 16 H20 GPUs, with Adam as the outer-loop optimizer, meta-learning rate $\beta = 2 \times 10^{-4}$, and inner-loop learning rate

$\alpha = 1 \times 10^{-1}$. For the boosting SOTA experiments, models are trained for 200k steps on 16 H20 GPUs, using Adam with meta-learning rate $\beta = 1 \times 10^{-4}$ and inner-loop learning rate $\alpha = 5 \times 10^{-2}$. In both settings, the outer-loop learning rate follows a cosine annealing schedule.

Dynamic resolution and batching. Following WorldMirror (Liu et al., 2025), we use dynamic image resolutions, maintaining total pixel counts between 100,000 and 250,000 per image, while sampling aspect ratios from 0.5 to 2.0. We also adopt dynamic batch sizing: for each GPU, the total number of support and query views per task is fixed at 24.

Meta-learning details. Geometry tasks adapt the selected encoder attention/FFN subset ϕ_{enc}, while appearance tasks adapt decoder convolutions ϕ_{dec}. These are existing network parameters, accounting for 8.6% and 2.8% of model parameters, respectively; they add neither a new module nor a per-scene parameter bank. During training, the inner-loop step count M is randomly sampled from $\{3, 5, 7, 10\}$ at each meta-iteration. This randomization prevents the meta-parameters from overfitting to a fixed adaptation depth and encourages a stable adaptation trajectory across different values of M. At inference, we fix $M = 5$.

Target-domain calibration and hyperparameters. In the main experiments, k counts the distinct support scenes whose labeled views form a target-domain calibration episode. These scenes belong to the same target domain as the evaluation scenes and are disjoint from them; the default $k = 1$ episode uses a single support scene. The MatrixCity and DL3DV evaluations in Tables 1 and 6, respectively, use 128 test scenes with four views per scene in each domain. FT-k starts from Static-Sup; *MetaFeed*-k starts from the meta-trained initialization. They use identical target support, adaptable parameter subsets, and evaluation protocols. Calibration adapts ϕ using M inner-loop steps on the support episode once per target domain; the resulting weights are then fixed for ordinary feed-forward inference on held-out scenes. The support signal is supervised: it may consist of independent ground truth or teacher predictions, as specified for each experiment. Hyperparameters were selected once in preliminary validation for stability and efficiency, then fixed across targets: $(\alpha, K, M) = (0.1, 1, 5)$ for controlled experiments and $(0.05, 1, 5)$ for VGGT. The sensitivity study in Appendix E.4 varies these adaptation settings and reports five-seed statistics with fixed test scenes and matched support draws.

D DL3DV Pseudo-Targets and Evaluation

DL3DV (Ling et al., 2024) provides high-quality multi-view captures of real-world outdoor scenes but does not include ground-truth labels for material attributes. We generate pseudo-labels for albedo, roughness, metallic, and surface normal as follows. These targets define an auxiliary teacher-fidelity/domain-calibration study, not an independent test of material accuracy. All methods use the same MVInverse targets, and support scenes are disjoint from the 128 evaluation scenes. Material-accuracy claims instead rely on datasets with independent ground truth.

Scene filtering. We first exclude scenes whose dominant content is transparent or highly reflective, since inverse-rendering predictions are unreliable in such regions and would inject systematic noise into the pseudo-labels. We additionally exclude scenes that are visually ambiguous, dominated by distant views, or contain visible reconstruction artifacts, as these conditions tend to produce unstable or low-confidence pseudo-labels.

View sampling and prediction. For each retained scene, we randomly sample views and run MVInverse (Wu et al., 2025) to obtain per-view predicted albedo, roughness, metallic, and surface normal maps. MVInverse is chosen because it operates in a feed-forward manner over multi-view inputs and produces spatially consistent material predictions across views of the same scene.

Valid-region masking. Outdoor scenes contain substantial regions where material prediction is ill-defined, most notably the sky. We use SAM3 (Carion et al., 2025) to segment sky and other invalid regions in each sampled view, and convert the resulting masks into per-view valid masks. Pseudo-label supervision is applied only within these valid masks during evaluation, and invalid pixels are excluded from all reported metrics on DL3DV.

Table 6: **Auxiliary teacher agreement on DL3DV.** Metrics compare predictions with MVInverse pseudo-targets, rather than independent material ground truth. *FT-k* and *MetaFeed-k* adapt the same decoder parameters using the same k support scenes, disjoint from the 128 evaluation scenes. Best per column is highlighted in **bold**.

Method	Albedo			Roughness	Metallic	Normal	
	PSNR↑	SSIM↑	LPIPS↓	RMSE↓	RMSE↓	Mean↓	Median↓
Static-Sup	14.43	0.734	0.314	0.184	0.171	19.23	13.27
FT-1	14.36	0.733	0.420	0.285	0.168	23.77	18.30
FT-3	15.87	0.753	0.358	0.213	0.165	26.95	20.93
FT-5	15.74	0.749	0.383	0.166	0.183	20.29	15.51
MetaFeed-1	17.05	0.817	0.256	0.070	**0.124**	19.87	13.74
MetaFeed-3	17.87	0.839	0.268	**0.064**	0.133	19.19	13.32
MetaFeed-5	**18.18**	**0.846**	**0.242**	0.073	0.172	**18.82**	**12.74**

Teacher-agreement results. Table 6 reports albedo PSNR/SSIM/LPIPS, roughness and metallic RMSE, and normal mean/median angular error against MVInverse pseudo-targets. *MetaFeed* improves agreement with the teacher over matched fine-tuning on most reported metrics. These results characterize teacher-supported calibration on real imagery; independent material accuracy is evaluated on datasets with ground truth.

E Additional Experimental Results

We examine external appearance references, same-data geometry controls, and support supervision. We also report adaptation sensitivity and support-sampling statistics. Appearance accuracy is evaluated against independent ground truth on MatrixCity and Structured3D; DL3DV results measure agreement with MVInverse.

E.1 External References and Distillation

Table 7 compares direct MVInverse (Wu et al., 2025) and RGB↔X (Zeng et al., 2024) predictions, ordinary distillation from Static-Sup, and *MetaFeed* with ground-truth or MVInverse-labeled support. MatrixCity and Structured3D are included in MVInverse's training sources, so its direct predictions are an external reference rather than a matched out-of-domain comparison. A direct MVInverse row on DL3DV would instead evaluate the teacher against its own targets and is therefore uninformative.

The distilled model uses pseudo-labels for the full MatrixCity training set. It nearly matches ground-truth-supported *MetaFeed* in MatrixCity albedo and has lower roughness and metallic RMSE than either *MetaFeed* support variant. However, the same distilled model obtains 14.33 PSNR on unseen Structured3D, compared with 14.63 before distillation and 17.86/17.29 for *MetaFeed* with one ground-truth/teacher-labeled support scene from that target. These rows compare full target-specific retraining with one-scene re-adaptation; they do not match the amount of target supervision. The objective is reusable cross-domain calibration, not replacement of a more accurate teacher.

E.2 Geometry Controls and Parameter-Efficient Adaptation

The same-data VGGT-FT and VGGT-FT-1 controls in Table 3 use the same VGGT architecture and nine synthetic datasets as *MetaFeed*. VGGT-FT is competitive on NRGBD but degrades relative to the public VGGT checkpoint on 7-Scenes and DTU. Relative to matched supervised training with VGGT-FT, uncalibrated *MetaFeed* reduces CD on all three benchmarks. Calibration using a single support episode from a separate scene within the same target domain further reduces CD on disjoint evaluation scenes, separating the benefit of the initialization from that of target adaptation.

We also apply rank-8 LoRA (Hu et al., 2021) to the same encoder $Q/K/V$ projections and use the same support and evaluation protocol. Table 8 compares 5, 50, and 100 adaptation steps. At five steps, LoRA uses less time and memory than the original parameter subset but yields higher CD than *MetaFeed*-1 on all three benchmarks. Increasing LoRA adaptation to 50 or 100 steps worsens CD

Table 7: **External appearance references and target-specific distillation.** Evaluation uses independent ground truth. Full MatrixCity distillation and one-scene calibration use different amounts of target-domain supervision. MVInverse is reported as a teacher reference.

Method	Target-domain supervision	MatrixCity			Structured3D
		Albedo PSNR↑	Roughness RMSE↓	Metallic RMSE↓	Albedo PSNR↑
MVInverse (direct)	None at evaluation	20.38	0.138	0.087	22.15
RGB↔X (Zeng et al., 2024)	None at evaluation	13.84	0.396	0.253	17.08
Static-Sup	None	12.70	0.623	0.220	14.63
Static-Sup + MVInverse distillation	Full MatrixCity-train pseudo-labels	18.15	0.151	0.092	14.33
MetaFeed-1	One GT support scene per target	18.27	0.173	0.109	17.86
MetaFeed-1	One MVInverse-labeled support scene per target	16.35	0.191	0.130	17.29

Table 8: **Parameter-efficient geometry adaptation.** Rank-8 LoRA is applied to the encoder $Q/K/V$ projections using the same support and evaluation protocol. We report runtime, peak memory, and reconstruction CD.

Method	Steps	Time (s)↓	Memory (GB)↓	7-Scenes CD↓	NRGBD CD↓	DTU CD↓
VGGT-FT-1	5	1.533	14.53	0.059	0.049	1.876
VGGT-FT + LoRA	5	1.021	6.68	0.056	0.046	1.798
VGGT-FT + LoRA	50	10.76	6.68	0.060	0.047	1.856
VGGT-FT + LoRA	100	21.32	6.68	0.064	0.048	1.915
MetaFeed-1	5	1.533	14.53	0.040	0.043	1.018

rather than closing this gap. This is an accuracy–resource trade-off under the reported configurations, not a claim of equal trainable parameter counts or equal total computation.

E.3 Support Supervision

Table 9 evaluates sensitivity to noisy geometry support. Increasing the reported noise level from 0 to 0.10 raises CD from 0.161 to 0.185, while remaining below the unadapted Static-Sup value of 0.298. The experiment supports tolerance to the tested label perturbations, rather than independence from support quality.

Table 10 compares ground-truth and appearance support generated by DiffusionRenderer (Liang et al., 2025). Teacher-supported *MetaFeed* obtains 17.39 PSNR versus 17.86 with ground-truth support, with 0.52 s calibration and 0.19 s per query in this experiment. Direct DiffusionRenderer prediction is more accurate at 20.61 PSNR but requires more than 30 s per query. Teacher support thus provides a way to transfer a supervised signal into fast repeated inference, with an accuracy trade-off. The MVInverse-support results in Table 7 give a second teacher-based comparison. Neither experiment removes the requirement for supervised calibration.

E.4 Support Sampling and Adaptation Hyperparameters

Table 11 reports mean and standard deviation over five seeds with fixed test scenes and matched support draws. The sweeps cover support size K, inner-loop steps M, and learning rate α. MatrixCity albedo PSNR increases from 18.27 ± 0.12 at $K = 1$ to 18.55 ± 0.08 at $K = 20$, while 7-Scenes CD decreases from 0.040 ± 0.004 to 0.036 ± 0.002. More support yields modest gains in these statistics; one-scene calibration remains the default. For the step sweep, increasing adaptation from one to five steps raises MatrixCity PSNR from 18.05 ± 0.13 to 18.27 ± 0.12 and lowers 7-Scenes CD from 0.044 ± 0.005 to 0.040 ± 0.004. Small differences among nearby step counts support selecting the default for stability and efficiency rather than tuning it separately for each target.

Table 9: **Geometry adaptation with noisy support labels.** We compare clean and noisy ground-truth support against the unadapted supervised baseline.

Method	Support	Noise	Accuracy↓	Completeness↓	CD↓
Static-Sup	None	N/A	0.130	0.467	0.298
MetaFeed-1	Clean GT	0	0.113	0.210	0.161
MetaFeed-1	Noisy GT	0.02	0.113	0.215	0.165
MetaFeed-1	Noisy GT	0.05	0.115	0.227	0.172
MetaFeed-1	Noisy GT	0.10	0.122	0.242	0.182

Table 10: **Appearance adaptation with teacher-generated support.** We separate one-time adaptation from per-query inference and include direct teacher predictions as a reference.

Method	Support source	PSNR↑	SSIM↑	LPIPS↓	Adaptation (s)↓	Query (s)↓
MetaFeed-1	Ground truth	17.86	0.756	0.231	0.52	0.19
MetaFeed-1	DiffusionRenderer	17.39	0.733	0.243	0.52	0.19
DiffusionRenderer (Liang et al., 2025)	N/A	20.61	0.824	0.171	N/A	> 30

Modality-specific adaptation curves. Figure 6(a–c) reports modality-specific appearance results at $\alpha = 0.1$ and one-shot support; panel (d) shows metallic error under varying learning rates. The MatrixCity albedo value 12.29 shown at zero in panel (a) is a Static-Sup reference, not the unadapted *MetaFeed* initialization. Among the adapted settings in the figure, MatrixCity albedo peaks at five steps, while roughness and metallic trends differ across datasets. These curves and the five-seed table are reported separately; the plotted values are not presented as the five-seed means. DL3DV curves quantify teacher agreement throughout.

Effect of support-set size. Figure 7 visualizes the K sweep in Table 11, with error bars showing one standard deviation across five seeds. Increasing support size from $K = 1$ to $K = 20$ raises MatrixCity albedo PSNR from 18.27 to 18.55 and lowers 7-Scenes CD from 0.040 to 0.036. The mean gains become smaller as K increases, while one-scene calibration remains a compact default.

F Limitations

The adaptability of *MetaFeed* remains bounded by the diversity of meta-training tasks. A target domain far from that distribution may not admit effective adaptation within a few steps. Broader task distributions and meta-regularization are possible ways to address this limitation (Volpi et al., 2021; Gao et al., 2022; Wang et al., 2025a).

The framework also requires supervised target-domain support. Dense geometry labels can be costly, and physical material ground truth is generally unavailable for real captures. Teacher predictions can supply an adaptation signal, but introduce teacher error and do not establish independent material accuracy on DL3DV. Direct teacher inference may be more accurate, as demonstrated by DiffusionRenderer; the practical benefit of teacher-supported calibration is fast subsequent queries. Self-supervised inner-loop objectives based on view-synthesis or geometric consistency remain future work (Liu et al., 2023; Chen et al., 2025).

Finally, meta-training is more expensive than standard supervision, and entering a new target domain incurs a one-time calibration cost. These costs are reported in Appendix G; they are not increases in persistent network capacity.

G Efficiency Analysis

G.1 Training Cost

Table 12 reports throughput, total wall-clock time, relative step time, and peak memory under the same controlled training setup. Standard supervision takes 21 hours, compared with 32 hours for Meta-DPT

Table 11: **Sensitivity to support size and adaptation hyperparameters.** We report mean ± standard deviation over five seeds with fixed test scenes and matched support draws.

Sweep	Value	MatrixCity PSNR↑	7-Scenes CD↓
K	1	18.27 ± 0.12	0.040 ± 0.004
	5	18.46 ± 0.10	0.038 ± 0.003
	10	18.52 ± 0.09	0.037 ± 0.003
	20	18.55 ± 0.08	0.036 ± 0.002
M	1	18.05 ± 0.13	0.044 ± 0.005
	3	18.21 ± 0.12	0.041 ± 0.004
	5	18.27 ± 0.12	0.040 ± 0.004
	7	18.29 ± 0.13	0.039 ± 0.005
	9	18.26 ± 0.14	0.041 ± 0.006
α	0.001	17.85 ± 0.15	0.046 ± 0.006
	0.005	18.08 ± 0.13	0.044 ± 0.005
	0.01	18.19 ± 0.12	0.042 ± 0.004
	0.05	18.25 ± 0.12	0.041 ± 0.004
	0.1	18.27 ± 0.12	0.040 ± 0.004
	0.5	18.12 ± 0.18	0.043 ± 0.007

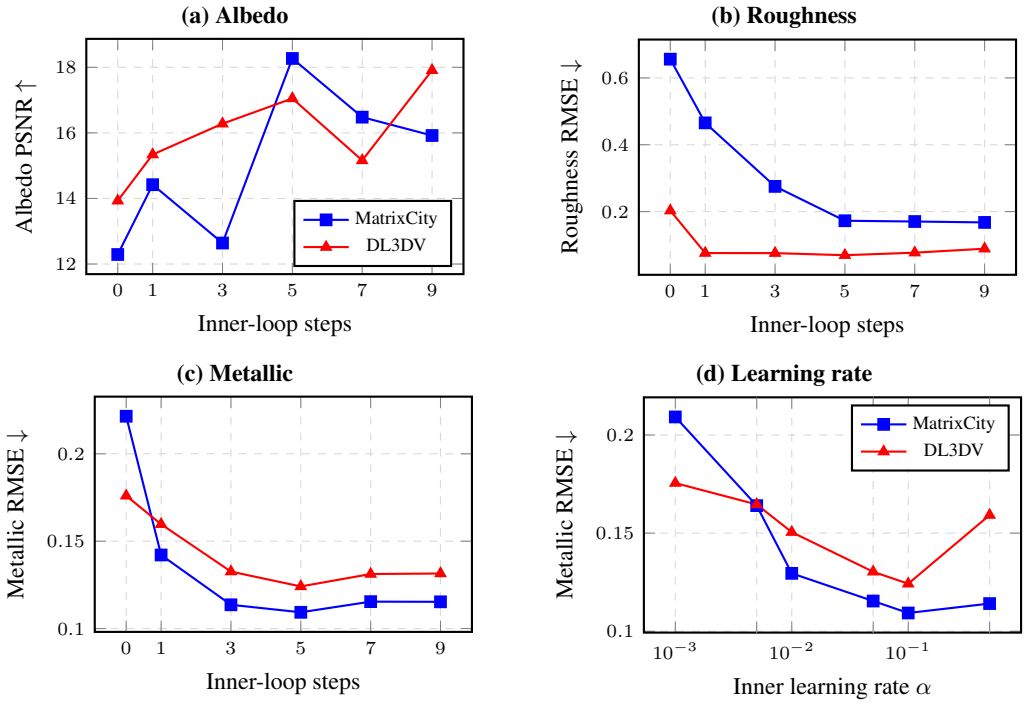


Figure 6: **Sensitivity to adaptation steps and learning rate.** Panels (a)–(c) vary the inner-loop step count at $\alpha = 0.1$ with one support scene; panel (d) varies α for metallic prediction. The best step count depends on the output and dataset; the learning-rate curves attain their lowest error at $\alpha = 0.1$ among the tested values. In (a), the MatrixCity point at zero is a Static-Sup reference, not an unadapted *MetaFeed* result. DL3DV results measure teacher agreement.

and 37 hours for Meta-Backbone. The corresponding per-step overheads are approximately $1.5\times$ and $1.7\times$. Only the selected 2.8% or 8.6% parameter subset is adapted in the inner loop, but support optimization still adds training computation and memory. These measurements characterize training cost, rather than an accuracy comparison at equal total computation.

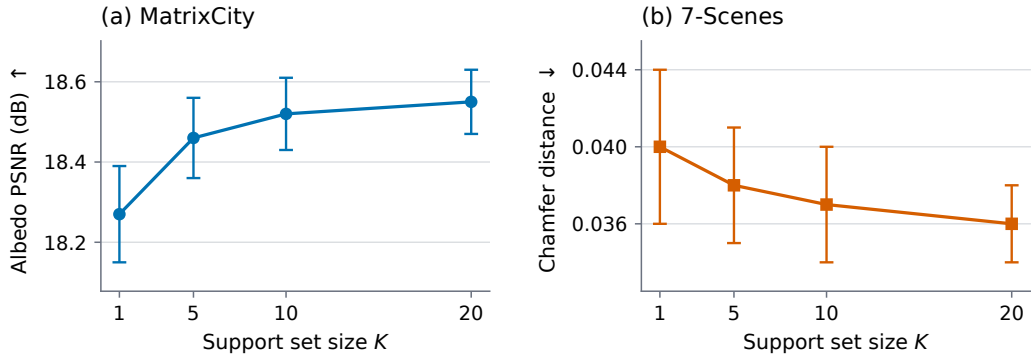


Figure 7: **Effect of K-shot support size.** Mean performance and error bars (one standard deviation) over five seeds, with fixed test scenes and matched support draws.

Table 12: **Training cost.** Throughput, total wall-clock time, relative step time, and peak memory are measured under the same controlled training setup.

Method	Throughput (steps/s)↑	Total time↓	Relative step time↓	Peak memory (GB)↓
No-Meta	0.26	21 h	1.0×	≈ 30
Meta-DPT	0.17	32 h	1.5×	≈ 55
Meta-Backbone	0.15	37 h	1.7×	≈ 55

G.2 Calibration and Query Inference

Table 13 reports runtime and peak GPU memory for the indicated support/query configurations. No-Meta is an ordinary forward pass; Meta-Backbone adapts ϕ_{enc} and Meta-DPT adapts ϕ_{dec} before the query pass. All measurements in this table use a single NVIDIA H20 GPU and five inner-loop steps. The support counts in this timing table denote views, separately from the scene counts used to define the main k-shot evaluation.

Results. At five query views and one support view, No-Meta requires 0.237 s and 7.42 GB. Meta-DPT requires 0.593 s and 11.15 GB, an increment of 0.356 s and 3.73 GB; Meta-Backbone requires 1.533 s and 14.53 GB. Meta-DPT time rises to 1.818 s at ten support views, while Meta-Backbone peak memory is approximately 25 GB from three support views onward. The higher calibration memory reflects temporary gradients, activations, and adapted weights, not new persistent model parameters.

Calibration is performed only once per target domain. Subsequent queries use the fixed adapted weights in an ordinary forward pass without additional optimization. In the separate appearance support experiment of Table 10, calibration takes 0.52 s and each query takes 0.19 s for both ground-truth and teacher-generated support; direct DiffusionRenderer inference takes more than 30 s per query. These measurements distinguish one-time adaptation from repeated query cost and are reported separately from the multi-view timings above.

Table 13: **Calibration and inference efficiency.** Runtime (seconds) and peak GPU memory (GB) for the indicated query and support views, with five adaptation steps for Meta-DPT and Meta-Backbone. No-Meta uses an ordinary forward pass. Calibration is performed once per target domain, after which the adapted model uses ordinary feed-forward inference.

Setting	**No-Meta**		**Meta-DPT** (ϕ_{dec})		**Meta-Backbone** (ϕ_{enc})	
	Time (s)	Mem (GB)	Time (s)	Mem (GB)	Time (s)	Mem (GB)
5 query, 1 support	0.237	7.42	0.593	11.15	1.533	14.53
5 query, 3 support	0.237	7.42	0.806	11.23	2.378	25.06
5 query, 5 support	0.237	7.42	1.085	11.31	2.390	25.09
5 query, 7 support	0.237	7.42	1.346	11.39	2.409	25.10
5 query, 10 support	0.237	7.42	1.818	12.22	2.376	25.12
3 query, 1 support	0.148	7.09	0.467	9.47	1.289	14.46
3 query, 3 support	0.147	7.09	0.705	9.55	2.176	25.02
3 query, 5 support	0.151	7.09	0.997	9.63	2.190	25.05
1 query, 1 support	0.081	6.76	0.394	7.81	1.164	14.39
1 query, 3 support	0.076	6.76	0.631	8.50	2.038	24.92
1 query, 5 support	0.077	6.76	0.903	9.35	2.062	24.96