Appendix A Preliminary

A.1 Camera Parameters Solver

To facilitate pose estimation and validate the implicit representation of camera parameters within our features, we introduce a specialized solver. Following the camera head of VGGT [9], our pose solver involves using a frozen DPT head [52] to compute a 3D point cloud. The 3D points (X, Y, Z) are then projected onto the 2D camera plane as (u, v):

$$\begin{bmatrix} u \\ v \end{bmatrix} = f \cdot \frac{1}{Z+s} \cdot \begin{bmatrix} X \\ Y \end{bmatrix} \tag{A1}$$

f is focal length, s is depth shift. We aim to minimize the objective function presented in Eq. (A2).

$$\min_{f,s} \sum_{i=1}^{N} \left\| \frac{\mathbf{XY}_i}{Z_i + s} - \mathbf{uv}_i \right\|^2 \tag{A2}$$

$XY_i = (x_i, y_i)$ is the 2D coordinates of the i-th point, $uv_i = (u_i, v_i)$ is the observed 2D pixel coordinates of the i-th point. Z_i is the depth value of the i-th point, N means the total number of points. Followed by Moge, a solver is employed to iteratively estimate the focal length f and depth shift s, as formulated in Eq. (A3).

$$\begin{cases} f^\star &= \frac{\sum_{i=1}^{N} \frac{xy_i}{z_i + s^\star \cdot uv_i}}{\sum_{i=1}^{N} ||\frac{xy_i}{z_i + s^\star}||^2} \\ s^\star &= LM(\underset{s}{arg\min} ||r(s)||^2) \end{cases} \tag{A3}$$

To solve for f and s simultaneously, we employ an alternating optimization strategy. By first fixing s, the subproblem of solving for f becomes convex, yielding the following optimal solution:

$$f^\star(s) = \frac{\sum_{i=1}^{N} XY_i' \cdot uv_i}{\sum_{i=1}^{N} \|XY_i'\|^2} \tag{A4}$$

Here, $XY_i' = \frac{XY_i}{Z_i + s}$. The subsequent step involves back-substituting the expression for $f^\star(s)$ into the objective function to obtain a residual function that is a function of s alone, as shown in Eq. (A5).

$$\mathbf{r}(s) = \begin{bmatrix} f^\star(s) \cdot \mathbf{XY}_1' - \mathbf{uv}_1 \\ f^\star(s) \cdot \mathbf{XY}_2' - \mathbf{uv}_2 \\ \vdots \\ f^\star(s) \cdot \mathbf{XY}_N' - \mathbf{uv}_N \end{bmatrix} \tag{A5}$$

The final s is estimated via the Levenberg-Marquardt algorithm, as formulated in Eq. (A3), from which the camera focal length $f = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$ is subsequently derived. LM denotes the Levenberg-Marquardt solver, and z represents our depth prior.

The camera Field-of-View (FOV) sought in this work can be analytically derived according to Eq. (A6).

$$\begin{cases} FOV_x & = 2 \cdot arctan(\frac{W}{2 \cdot f_x}) \\ FOV_y & = 2 \cdot arctan(\frac{H}{2 \cdot f_y}) \end{cases} \tag{A6}$$

Here, W and H denote the width and height of the input image, respectively.

A.2 Relationship between Disparity and Depth

In stereoscopic vision, the three-dimensional structure of the world is reconstructed from a pair of two-dimensional images. The fundamental cue for this process is binocular disparity—the horizontal positional difference between the projections of a scene point in the left and right images. Mathematically, for a 3D world point $P(X, Y, Z)$, if its projections onto the left and right image planes have coordinates x_l and x_r respectively, the disparity d is defined as $d = x_l - x_r$. This disparity d is not a direct measure of distance but is geometrically inversely proportional to metric depth Z, the absolute distance from the observer, as described by the standard binocular imaging model: $Z = \frac{f \times B}{d}$, where f is the focal length and B is the baseline distance between the cameras. This equation provides a precise, quantitative mapping from image coordinates to a scaled metric interpretation of the scene, which is crucial for applications like robotic navigation and precise 3D reconstruction.

A.3 Definition of Training-free

The term training-free denotes a specific methodological paradigm wherein a pre-trained model is adapted or utilized for a downstream task without performing any gradient-based updates to its parameters [51, 70]. Our application of an entropy-based supervisory signal for data-free model merging squarely falls within this category. We leverage the intrinsic property that a model's predictive entropy—a measure of uncertainty calculated directly from its output distribution—can serve as a proxy objective to guide the merging of parameters from homologous models without any exposure to new labeled or unlabeled data . By minimizing the entropy of the merged model's predictions, the process encourages consensus and confidence in the unified output space, effectively resolving parameter-level interference based on a signal that is generated entirely from withinthe constituent models themselves, thereby satisfying the core criteria of being both data-free and reliant on an objective that does not require a separate training loop.

Appendix B Supplementary Experiments

B.1 Camera Pose Estimation

We posit that learning an implicit understanding of camera parameters is crucial for effective feature representation in stereo vision. To evaluate the model's awareness of camera geometry, we utilize the solver described in Appendix Sec. A.1 to estimate the Camera Field-of-View (FOV). Following the experimental setup described in Sec. 4.3, we conduct a comparative analysis between our framework and the native

FOV estimation capabilities of SOTA MDE and 3R models. As summarized in Tab. B1, our approach demonstrates a substantial margin of improvement across all four evaluation metrics. Specifically, StereoVGGT outperforms the second-best performing method by 69.5%, 59.8%, 60.8%, and 37.4%, respectively. These findings indicate that StereoVGGT possesses a more potent implicit representation of camera parameters than the original VGGT baseline. This enhanced camera knowledge serves as a robust foundation, providing the necessary priors for StereoVGGT to achieve superior stereo-vision feature representations.

Methods	FOV x ↓		FOV y ↓	
	med.	mean	med.	mean
fastVGGT	<u>12.71</u>	<u>16.58</u>	<u>6.83</u>	<u>8.13</u>
VGGT	12.85	16.60	7.08	<u>8.13</u>
DAv2 †	78.41	78.99	64.24	69.18
Moge-2 †	32.29	39.52	21.79	20.02
VGGT †	15.77	19.58	12.01	12.33
StereoVGGT (Ours)	**3.88**	**6.67**	**2.68**	**5.09**

Table B1 Evaluation of left-view camera Field of View (FOV) in degrees. The dataset evaluated in this study was ETH3D binocular stereo datasets. Each method receives only the left-view image as input and outputs the intrinsic parameters of the left camera. † indicate the methods utilize the identical camera-pose solver as that integrated into StereoVGGT. **Bold**: Best. <u>Underline</u>: Second Best.

B.2 Generalization to Other Frameworks

To further evaluate the generalization of StereoVGGT, we conducted backbone substitution experiments within the BridgeDepth framework [71], as shown in Tab. B2. BridgeDepth, a SOTA model for the Scene Flow dataset [72], is designed to support the seamless integration of frozen monocular depth estimation models. In our experiments, we replaced the default DepthAnything V2 [8] backbone with Moge-2 [10], VGGT [9], and our proposed StereoVGGT. The empirical results demonstrate that StereoVGGT consistently remains the most effective variant among the evaluated configurations, further substantiating its robustness across different stereo matching pipelines.

B.3 Computational Cost

This section compares the computational cost across the different models. All experiments were conducted on a single NVIDIA RTX 3090 GPU. The model inputs consisted of the left-view images from the KITTI dataset, along with the focal length and baseline length from the KITTI camera parameters. The output was a disparity map, and the results are presented in Tab. B3. The time reported in the table represents the average duration over 50 inference runs.

Baseline Network	Type	Backbone	D1 ↓	EPE ↓
IGEV-Stereo [13]	VFM	MobileNet V2 (original version)	5.3	0.47
		EfficientNet V2 † [25]	5.2	0.46
	3R	VGGT † [9]	5.3	0.46
	MDE	Moge-2 † [10]	5.1	0.45
		DepthAnything V2 † [8]	5.1	0.44
		StereoVGGT (Ours)	4.9	0.43
BridgeDepth [71]	MDE	DepthAnything V2 (original version)	<u>3.7</u>	<u>0.37</u>
		Moge-2 †	<u>3.7</u>	0.39
	3R	VGGT †	4.0	0.42
		StereoVGGT (Ours)	**3.3**	**0.33**

Table B2 Quantitative evaluation on Scene Flow test set. The error threshold is 1 px. † denotes our reproduced version, for which no publicly available source code exists. Light Gray shading indicates the baseline method.

Despite incorporating a substantial number of parameters, StereoVGGT does not suffer from a significant increase in inference time, as a large portion of the parameters from the MDE model and VGGT are loaded but remain inactive during inference. Remarkably, StereoVGGT achieves a faster inference speed than DepthAnything V2. The minimal increase in inference time, despite the high computational complexity, is likely due to the architecture's efficient memory access patterns and uniform structure, which reduces bandwidth bottlenecks and enhances parallelism on GPUs.

Model	Time (s) ↓	TFLOPs ↓	Param. (M) ↓	EPE (px) ↓
Moge-2	0.156	1.50	326.21	6.74
DA v2	0.328	1.95	335.32	5.88
VGGT	0.197	3.78	1256.54	13.07
StereoVGGT	0.203	5.77	1891.31	2.71

Table B3 To evaluate computational cost, we performed inference for each model on the KITTI dataset using the same experimental setup as in Tab. 4.

B.4 Disparity Perception under Ill-posed Conditions

Ill-posed conditions refers to challenging scenarios such as thin structures, reflective surfaces, and textureless or weakly-textured regions, where achieving accurate disparity estimation is particularly difficult. The ETH3D dataset [65] is specifically designed to include these ill-posed conditions. Building upon the results shown in Tab. 4 and Tab. B1, this section offers a deeper analysis of the model's performance on this dataset.

In addition to the quantitative analysis provided in the main manuscript, we present a visual comparison of different types of ill-posed regions in Fig. B1. Specifically, MDE models, exemplified by DepthAnything V2 [8], frequently confuse the spatial relationships between scene objects and the imaging plane, whereas VGGT [9] often produces blurred object boundaries. In contrast, StereoVGGT achieves both fine-grained edge estimation and accurate relative distance estimation for objects.

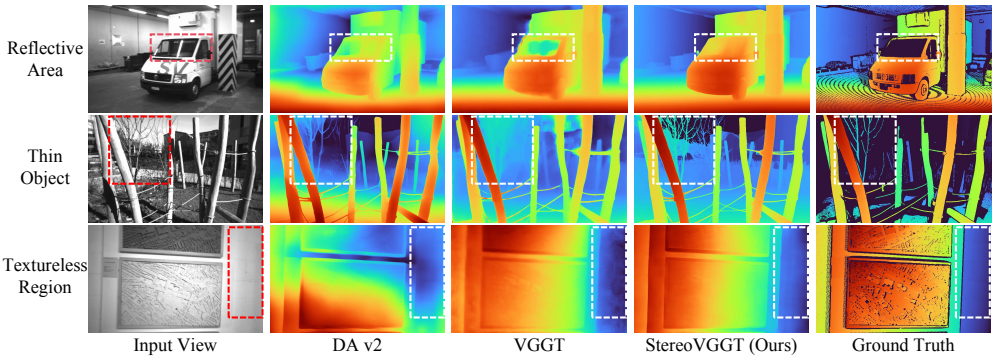
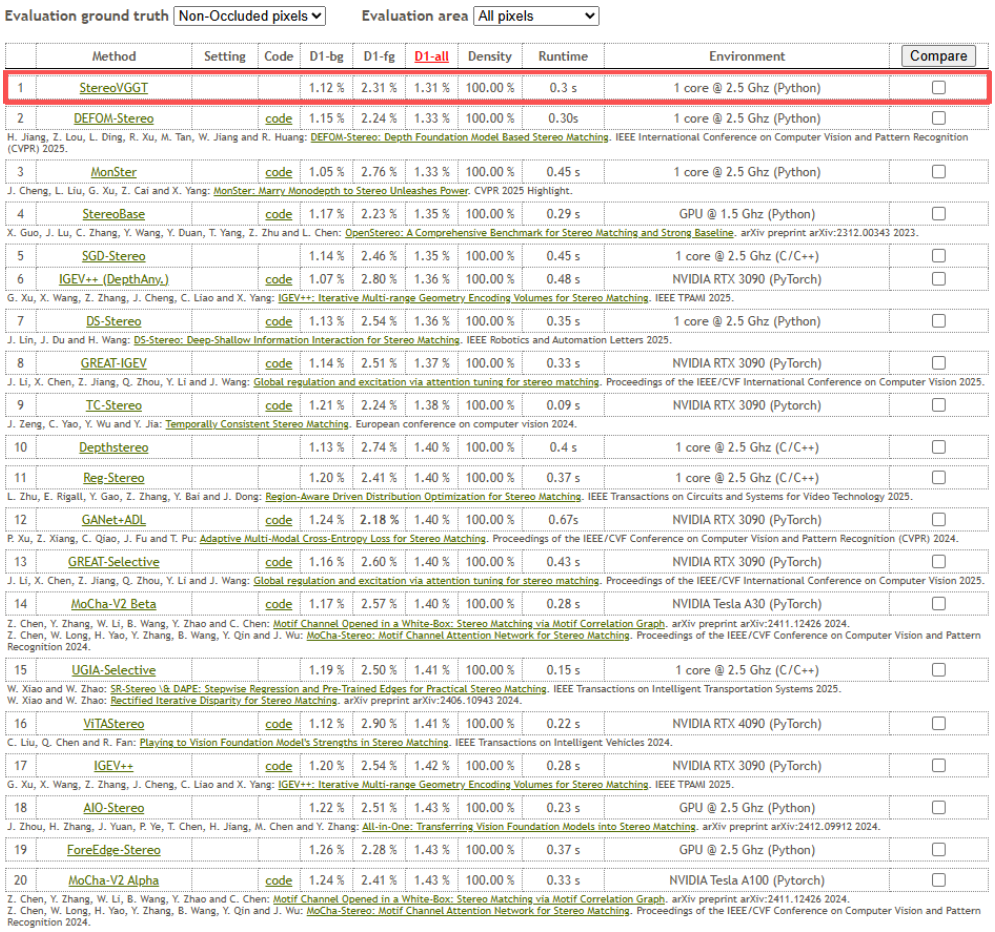


Fig. B1 Disparity map comparisons across state-of-the-art models on challenging ill-posed regions. The models use monocular condition input of binocular cameras, consisting of a left view, focal length, and baseline length. The red boxes and white boxes highlight the main differences.

B.5 Benchmark performance

By employing StereoVGGT as the feature backbone integrated with the IGEV decoder [13] (detailed in Sec. 3.5), our proposed stereo matching network achieved the top-ranking position on the prestigious KITTI online benchmark at the time of submission, as illustrated in Fig. B2.



Evaluation ground truth: Non-Occluded pixels Evaluation area: All pixels

	Method	Setting	Code	D1-bg	D1-fg	D1-all	Density	Runtime	Environment	Compare
1	StereoVGGT			1.12 %	2.31 %	1.31 %	100.00 %	0.3 s	1 core @ 2.5 Ghz (Python)	
2	DEFOM-Stereo		code	1.15 %	2.24 %	1.33 %	100.00 %	0.30s	1 core @ 2.5 Ghz (Python)	
	H. Jiang, Z. Lou, L. Ding, R. Xu, M. Tan, W. Jiang and R. Huang: DEFOM-Stereo: Depth Foundation Model Based Stereo Matching. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2025.									
3	MonSter		code	1.05 %	2.76 %	1.33 %	100.00 %	0.45 s	1 core @ 2.5 Ghz (Python)	
	J. Cheng, L. Liu, G. Xu, Z. Cai and X. Yang: MonSter: Marry Monodepth to Stereo Unleashes Power. CVPR 2025 Highlight.									
4	StereoBase		code	1.17 %	2.23 %	1.35 %	100.00 %	0.29 s	GPU @ 1.5 Ghz (Python)	
	X. Guo, J. Lu, C. Zhang, Y. Wang, Y. Duan, T. Yang, Z. Zhu and L. Chen: OpenStereo: A Comprehensive Benchmark for Stereo Matching and Strong Baseline. arXiv preprint arXiv:2312.00343 2023.									
5	SGD-Stereo			1.14 %	2.46 %	1.35 %	100.00 %	0.45 s	1 core @ 2.5 Ghz (C/C++)	
6	IGEV++ (DepthAny.)		code	1.07 %	2.80 %	1.36 %	100.00 %	0.48 s	NVIDIA RTX 3090 (PyTorch)	
	G. Xu, X. Wang, Z. Zhang, J. Cheng, C. Liao and X. Yang: IGEV++: Iterative Multi-range Geometry Encoding Volumes for Stereo Matching. IEEE TPAMI 2025.									
7	DS-Stereo		code	1.13 %	2.54 %	1.36 %	100.00 %	0.35 s	1 core @ 2.5 Ghz (Python)	
	J. Lin, J. Du and H. Wang: DS-Stereo: Deep-Shallow Information Interaction for Stereo Matching. IEEE Robotics and Automation Letters 2025.									
8	GREAT-IGEV		code	1.14 %	2.51 %	1.37 %	100.00 %	0.33 s	NVIDIA RTX 3090 (PyTorch)	
	J. Li, X. Chen, Z. Jiang, Q. Zhou, Y. Li and J. Wang: Global regulation and excitation via attention tuning for stereo matching. Proceedings of the IEEE/CVF International Conference on Computer Vision 2025.									
9	TC-Stereo		code	1.21 %	2.24 %	1.38 %	100.00 %	0.09 s	NVIDIA RTX 3090 (Pytorch)	
	J. Zeng, C. Yao, Y. Wu and Y. Jia: Temporally Consistent Stereo Matching. European conference on computer vision 2024.									
10	Depthstereo			1.13 %	2.74 %	1.40 %	100.00 %	0.4 s	1 core @ 2.5 Ghz (C/C++)	
11	Reg-Stereo			1.20 %	2.41 %	1.40 %	100.00 %	0.37 s	1 core @ 2.5 Ghz (C/C++)	
	L. Zhu, E. Rigall, Y. Gao, Z. Zhang, Y. Bai and J. Dong: Region-Aware Driven Distribution Optimization for Stereo Matching. IEEE Transactions on Circuits and Systems for Video Technology 2025.									
12	GANet+ADL		code	1.24 %	**2.18 %**	1.40 %	100.00 %	0.67s	NVIDIA RTX 3090 (PyTorch)	
	P. Xu, Z. Xiang, C. Qiao, J. Fu and T. Pu: Adaptive Multi-Modal Cross-Entropy Loss for Stereo Matching. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2024.									
13	GREAT-Selective		code	1.16 %	2.60 %	1.40 %	100.00 %	0.43 s	NVIDIA RTX 3090 (PyTorch)	
	J. Li, X. Chen, Z. Jiang, Q. Zhou, Y. Li and J. Wang: Global regulation and excitation via attention tuning for stereo matching. Proceedings of the IEEE/CVF International Conference on Computer Vision 2025.									
14	MoCha-V2 Beta		code	1.17 %	2.57 %	1.40 %	100.00 %	0.28 s	NVIDIA Tesla A30 (PyTorch)	
	Z. Chen, Y. Zhang, W. Li, B. Wang, Y. Zhao and C. Chen: Motif Channel Opened in a White-Box: Stereo Matching via Motif Correlation Graph. arXiv preprint arXiv:2411.12426 2024. Z. Chen, W. Long, H. Yao, Y. Zhang, B. Wang, Y. Qin and J. Wu: MoCha-Stereo: Motif Channel Attention Network for Stereo Matching. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2024.									
15	UGIA-Selective			1.19 %	2.50 %	1.41 %	100.00 %	0.15 s	1 core @ 2.5 Ghz (C/C++)	
	W. Xiao and W. Zhao: SR-Stereo \& DAPE: Stepwise Regression and Pre-Trained Edges for Practical Stereo Matching. IEEE Transactions on Intelligent Transportation Systems 2025. W. Xiao and W. Zhao: Rectified Iterative Disparity for Stereo Matching. arXiv preprint arXiv:2406.10943 2024.									
16	ViTAStereo		code	1.12 %	2.90 %	1.41 %	100.00 %	0.22 s	NVIDIA RTX 4090 (PyTorch)	
	C. Liu, Q. Chen and R. Fan: Playing to Vision Foundation Model's Strengths in Stereo Matching. IEEE Transactions on Intelligent Vehicles 2024.									
17	IGEV++		code	1.20 %	2.54 %	1.42 %	100.00 %	0.28 s	NVIDIA RTX 3090 (PyTorch)	
	G. Xu, X. Wang, Z. Zhang, J. Cheng, C. Liao and X. Yang: IGEV++: Iterative Multi-range Geometry Encoding Volumes for Stereo Matching. IEEE TPAMI 2025.									
18	AIO-Stereo			1.22 %	2.51 %	1.43 %	100.00 %	0.23 s	GPU @ 2.5 Ghz (Python)	
	J. Zhou, H. Zhang, J. Yuan, P. Ye, T. Chen, H. Jiang, M. Chen and Y. Zhang: All-in-One: Transferring Vision Foundation Models into Stereo Matching. arXiv preprint arXiv:2412.09912 2024.									
19	ForeEdge-Stereo			1.26 %	2.28 %	1.43 %	100.00 %	0.37 s	GPU @ 2.5 Ghz (Python)	
20	MoCha-V2 Alpha		code	1.24 %	2.41 %	1.43 %	100.00 %	0.33 s	NVIDIA Tesla A100 (Pytorch)	
	Z. Chen, Y. Zhang, W. Li, B. Wang, Y. Zhao and C. Chen: Motif Channel Opened in a White-Box: Stereo Matching via Motif Correlation Graph. arXiv preprint arXiv:2411.12426 2024. Z. Chen, W. Long, H. Yao, Y. Zhang, B. Wang, Y. Qin and J. Wu: MoCha-Stereo: Motif Channel Attention Network for Stereo Matching. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2024.									

Fig. B2 StereoVGGT ranked 1^{st} on the KITTI online benchmark among all submission methods.