EventHub: Data Factory for Generalizable Event-Based Stereo Networks without Active Sensors

Supplementary Material

This document reports additional material related to the CVPR paper "EventHub: Data Factory for Generalizable Event-Based Stereo Networks without Active Sensors".

- First, we present an extended description of our Novel View Synthesis (NVS) pipeline in Section 3, including details about the depth regularizers used to improve depth estimation (Sec. 6.1), and the novel voxel-based confidence $\mathbf{C}_{\text{Vsize}}$ (Sec. 6.2).
- Next, we include additional implementation details, in particular, regarding the global trajectory $\Omega(\tau)$ (Section 8.1), the datasets splits (Sec. 8.2), and the stereo model losses (Sec. 8.3).
- Finally, we present extensive qualitative results regarding both generated data from [18, 61, 75] using our EventHub pipeline, and disparity estimation from our trained event stereo networks, using the three evaluation datasets [7, 18, 84].

6. Method Overview: Additional Details

In this section, we include an extended description of our EventHub pipeline.

6.1. Depth Regularizers

To improve the quality of our NVS generation pipeline, we rely on a subset of the following regularization strategies:

- $\mathcal{L}_{N-\text{mean}}$ and $\mathcal{L}_{N-\text{med}}$: both losses encourage agreement between depth and normal renderings, obtained through mean and median aggregation, respectively [25];
- $\mathcal{L}_{\text{DAv2}}$ promotes consistency between the rendered depth and the monocular predictions from DepthAnythingV2 [73];
- $\mathcal{L}_{\text{asc}}$ encourages density to increase monotonically along the ray direction;
- $\mathcal{L}_{\text{sparse}}$ fosters depth regularization using COLMAP [57] sparse 3D points;
- $\mathcal{L}_{\text{MASt3R}}$ guides the depth regularization following MASt3R predictions.

Ablation study and metrics for NVS. To assess the contribution of each depth regularizer, we conducted an ablation experiment on ScanNet++ [75], which provides ground-truth depth. In particular, we selected a small dataset split and evaluated the contribution of each regularizer using two metrics for NVS image quality (*i.e.*, PSNR and SSIM) and two metrics for depth evaluation (*i.e.*, MAE and $\delta \leq \rho$):

- **Peak Signal-to-Noise Ratio (PSNR)**. For color images $\mathbf{I}$ and ground-truth $\mathbf{I}^*$, PSNR is defined based on the mean squared error (MSE):

$$\text{MSE} = \frac{1}{N}\sum_i (\mathbf{I}_i - \mathbf{I}_i^*)^2, \qquad \text{PSNR} = -10\log_{10}(\text{MSE}), \tag{7}$$

  where N is the number of pixels, and $\mathbf{I}_i$ and $\mathbf{I}_i^*$ are the RGB values of the i-th pixel in the rendered and ground-truth images, respectively. Higher PSNR indicates better agreement with the ground truth.
- **Structural Similarity Index (SSIM)**. SSIM measures similarity between predicted and ground-truth color images $\mathbf{I}$ and $\mathbf{I}^*$ by comparing local windows $\mathbf{X} \in \mathcal{N}_{\mathbf{I}}$ and $\mathbf{Y} \in \mathcal{N}_{\mathbf{I}^*}$:

$$\text{SSIM}(\mathbf{I}, \mathbf{I}^*) = \frac{1}{M}\sum_{\mathbf{X}\in\mathcal{N}_{\mathbf{I}}}\sum_{\mathbf{Y}\in\mathcal{N}_{\mathbf{I}^*}} \frac{(2\mu_{\mathbf{X}}\mu_{\mathbf{Y}} + C_1)(2\sigma_{\mathbf{XY}} + C_2)}{(\mu_{\mathbf{X}}^2 + \mu_{\mathbf{Y}}^2 + C_1)(\sigma_{\mathbf{X}}^2 + \sigma_{\mathbf{Y}}^2 + C_2)}, \tag{8}$$

  where M is the number of windows, C_1 and C_2 are constants, $\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}$ are local means, $\sigma_{\mathbf{X}}^2, \sigma_{\mathbf{Y}}^2$ the local variances, and $\sigma_{\mathbf{XY}}$ the local covariance. Higher values indicate better structural similarity.
- **Mean Absolute Error (MAE)**. It measures the average magnitude of errors:

$$\text{MAE} = \frac{1}{N}\sum_i |\mathbf{Z}_i - \mathbf{Z}_i^*|, \tag{9}$$

  where N is the number of pixels, $\mathbf{Z}_i$ and $\mathbf{Z}_i^*$ are the predicted and ground-truth depths of the i-th pixel, respectively.

Row	$\lambda_{N-\text{mean}}$	$\lambda_{N-\text{med}}$	λ_{asc}	λ_{sparse}	λ_{DAv2}	λ_{MASt3R}	PSNR	SSIM($\times$100)	MAE (cm)	$\delta \leq 1.25$ (%)
1	-	-	-	-	-	-	**33.85**	**87.36**	8.81	93.51
2	0.001	0.001	-	-	-	-	33.25	86.77	9.03	92.43
3	0.001	0.001	0.01	-	-	-	33.25	86.77	9.03	92.47
4	0.001	0.001	0.01	0.01	-	-	33.25	86.77	8.99	92.52
5	0.001	0.001	0.01	0.01	0.01	-	33.19	86.73	6.61	96.23
6	0.001	0.001	0.01	0.01	0.01	0.01	26.86	79.64	38.71	67.31
7	0.001	0.001	0.01	-	0.01	-	33.20	86.73	6.58	96.25
8	0.001	0.001	-	-	0.01	-	33.19	86.73	6.57	96.29
9	-	-	-	-	0.01	-	33.74	87.24	7.15	95.89
10	0.0005	0.0005	-	-	0.01	-	33.37	86.91	**6.38**	**96.44**

Table 6. **Depth regularization ablation.** PSNR values are given in decibels. SSIM values are multiplied by a 100 factor. MAE values are reported in centimeters.

Model	PSNR ↑	MAE (cm) ↓	$\delta \leq 1.25$ (%) ↑	Setup Time (min/scene) ↓	FPS ↑
Depth Anything v3 [40]	19.19	41.94	53.92	∼**1**	**165**
Instant-NGP [48]	29.21	24.68	82.88	∼8	5
3DGS [31]	32.51	22.36	76.23	∼20	**165**
SVRaster [59]	**33.37**	**6.38**	**96.44**	∼20	143

Table 7. **Comparison between different NVS engines.** SVRaster achieves the best trade-off between rendering quality, setup time and rendering speed.

- **Threshold Accuracy**. It reports the percentage of predicted depths within a threshold ρ (in our ablation experiment $\rho = 1.25$) indicating the proportion of accurate predictions:

$$\text{Accuracy} = \frac{1}{N}\sum_i \chi\left(\max\left(\frac{\mathbf{Z}_i}{\mathbf{Z}_i^*}, \frac{\mathbf{Z}_i^*}{\mathbf{Z}_i}\right) \leq \rho\right) = \frac{1}{N}\sum_i \chi\left(\delta \leq \rho\right) \tag{10}$$

where N is the number of pixels, $\mathbf{Z}_i$ and $\mathbf{Z}_i^*$ are the predicted and ground-truth depths of the i-th pixel, respectively, and $\chi(\cdot)$ is the indicator function.

Ablation Analysis. Table 6 reports the results of our study on depth-guided regularization terms. The left columns indicate the weights λ set for each regularizer, starting with the default values from [59]. Without any regularization (first row), SVRaster achieves solid PSNR and SSIM but exhibits a relatively large depth error (MAE = 8.81 cm). Introducing the first four regularizers – *i.e.*, $\mathcal{L}_{\text{N-mean}}$ and $\mathcal{L}_{\text{N-med}}$ (row 2), $\mathcal{L}_{\text{asc}}$ (row 3), and $\mathcal{L}_{\text{sparse}}$ (row 4) – yields no meaningful improvements, aside from a marginal SSIM gain. In contrast, incorporating the monocular prior from DepthAnythingV2 [73] (row 5) produces a substantial reduction in depth error (25% decrease in MAE) while preserving nearly unchanged image quality. Adding $\mathcal{L}_{\text{MASt3R}}$ on top of all other regularizers (row 6), however, severely degrades performance. Given the strong influence of $\mathcal{L}_{\text{DAv2}}$, we perform additional ablations where the remaining regularizers are removed one at a time (rows 7, 8, and 9). This analysis shows minor contribution from $\mathcal{L}_{\text{asc}}$ and $\mathcal{L}_{\text{sparse}}$, but disabling $\lambda_{N-\text{mean}}$ and $\lambda_{N-\text{med}}$ leads to worse results than those of row 5. Therefore, we reintroduce these two terms with halved weights (row 10), which yields the best overall depth performance. We adopt this last configuration as the final set of depth-regularization weights for our NVS pipeline.

Impact of the NVS engine. To support our choice of using SVRaster to render both proxy labels and event streams, we report a comparison with other state-of-the-art novel view synthesis approaches in Table 7. In addition to rendering quality, we also consider the setup time necessary to process each single scene before starting the rendering process, as well as the speed at which data is generated. Notably, Depth Anything v3 has the lowest setup time, as it directly predicts a 3DGS field in a feed-forward fashion rather than a per-scene optimization process. However, this speed is traded for a much lower rendering quality. Instant-NGP still requires a low setup time, yet features a very low rendering speed and sub-optimal rendering quality. Finally, although requiring the highest setup time, 3DGS and SVRaster yields the highest rendering quality: among the two, SVRaster shines thanks to the careful use of depth regularization.

6.2. Novel Voxel-based Confidence

Despite the added depth regularization, the resulting depth maps may still contain noticeable noise. To address this issue, [61] introduced a trinocular photometric loss:

$$\mathcal{L}_{\text{NS}} = \lambda_{\text{disp}} \cdot \eta(\mathbf{C}_{\text{AO}}; \mu_{\text{AO}}) \cdot \mathcal{L}_{\text{disp}} + \mathbf{M}_{\text{auto}} \cdot \lambda_{\text{3p}} \cdot (1 - \eta(\mathbf{C}_{\text{AO}}; \mu_{\text{AO}})) \cdot \mathcal{L}_{\text{3p}}, \tag{11}$$

Confidence	Threshold	MAE (cm)	$\delta \leq 1.25$ (%)	Density (%)
-	-	6.38	96.44	100.00
$\mathbf{C}_{\text{AO}}$	0.35	6.26	**96.60**	95.45
$\mathbf{C}_{\text{Vsize}}$	0.75	**6.23**	96.57	**97.42**
$\mathbf{C}_{\text{AO}}$	0.40	6.22	**96.66**	92.04
$\mathbf{C}_{\text{Vsize}}$	0.80	**6.15**	96.61	**95.56**
$\mathbf{C}_{\text{AO}}$	0.45	6.20	**96.72**	87.38
$\mathbf{C}_{\text{Vsize}}$	0.85	**6.03**	96.69	**91.63**

Table 8. **Confidence threshold study.** Comparison between ambient occlusion confidence $\mathbf{C}_{\text{AO}}$ [61] and our voxel-based confidence $\mathbf{C}_{\text{Vsize}}$ on ScanNet++ [75]. MAE values are reported in centimeters.

where $\mathcal{L}_{\text{disp}}$ is the disparity supervision loss with respect to the estimated disparity $\mathbf{D}_e$ (further details in Sec. 8.3), $\lambda_{\text{disp}} = 1.0$ and $\lambda_{\text{3p}} = 0.1$ are the loss weights set to the default values in [61], $\eta(\mathbf{C}_{\text{AO}}; \mu_{\text{AO}})$ is the truncation function that truncates confidence $\mathbf{C}_{\text{AO}}$ using the threshold $\mu_{\text{AO}} = 0.5$:

$$\eta(\mathbf{C}; \mu) = \begin{cases} 0 & \text{if } \mathbf{C} \leq \mu \\ \mathbf{C} & \text{otherwise} \end{cases}, \qquad \mathbf{C}_{\text{AO}} = \text{norm}\left(\sum_{i=1}^{N} T_i \alpha_i^2\right), \qquad \text{norm}(\mathbf{X}) = \frac{\mathbf{X} - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})}, \tag{12}$$

and given the three rendered images $\mathbf{I}_{LL}, \mathbf{I}_L, \mathbf{I}_R$ – where $\mathbf{I}_{LL}$ and $\mathbf{I}_R$ are rendered after applying respective stereo translations $(b\ 0\ 0)^\top$ and $(-b\ 0\ 0)^\top$ to the translation component $\mathbf{t}_\tau$ of the virtual trajectory $\Gamma(\tau)$ or $\Omega(\tau)$ – we can define the trinocular photometric loss $\mathcal{L}_{\text{3p}}$ as follow:

$$\mathcal{L}_{\text{3p}}(\mathbf{I}_{LL}, \mathbf{I}_L, \mathbf{I}_R) = \min\Big(\mathcal{L}_{\text{2p}}\big(\mathbf{I}_L, \mathcal{W}(\mathbf{I}_{LL}, \mathbf{D}_e)\big), \mathcal{L}_{\text{2p}}\big(\mathbf{I}_L, \mathcal{W}(\mathbf{I}_R, -\mathbf{D}_e)\big)\Big), \tag{13}$$

where $\mathcal{L}_{\text{2p}}$ is the standard photometric loss, $\mathcal{W}(\cdot, \cdot)$ is the backward warping function using the estimated disparity $\mathbf{D}_e$ from the event stereo model, and $\mathbf{M}_{\text{auto}}$ is the automasking term that removes untextured regions. The standard photometric loss $\mathcal{L}_{\text{2p}}$ and the automasking term $\mathbf{M}_{\text{auto}}$ are defined, respectively, as follow:

$$\mathcal{L}_{\text{2p}}(\mathbf{I}, \mathbf{I}^{\mathcal{W}}) = \beta \frac{1 - \text{SSIM}(\mathbf{I}, \mathbf{I}^{\mathcal{W}})}{2} + (1 - \beta)\left|\mathbf{I} - \mathbf{I}^{\mathcal{W}}\right|, \tag{14}$$

$$\mathbf{M}_{\text{auto}} = \chi\left(\min \mathcal{L}_{\text{3p}}\left(\mathcal{W}(\mathbf{I}_{LL}, \mathbf{D}_e), \mathbf{I}_L, \mathcal{W}(\mathbf{I}_R, -\mathbf{D}_e)\right) < \min \mathcal{L}_{\text{3p}}\left(\mathbf{I}_{LL}, \mathbf{I}_L, \mathbf{I}_R\right)\right). \tag{15}$$

We studied a replacement for $\mathbf{C}_{\text{AO}}$ that exploits the properties peculiar to the underlying NVS engine [59], and introduced $\mathbf{C}_{\text{Vsize}}$ using the voxel size as a confidence measure. Indeed, voxel sizes are defined during scene optimization and encouraged to be smaller for voxels seen from multiple viewpoints (*i.e.*, those points in the scene that are more constrained by multi-view geometry). With reference to Equation (3), we include additional details:

$$\mathbf{C}_{\text{Vsize}} = \text{norm}\left(\sum_{i=1}^{N} T_i s_i\right) \odot \text{norm}\left(\sum_{i=1}^{N} T_i \alpha_i\right) = \mathbf{C}'_{\text{Vsize}} \odot \mathbf{C}_{\text{hole}}, \tag{16}$$

where $\mathbf{C}'_{\text{Vsize}}$ returns high confidence to pixels whose rays intersect small voxels, and $\mathbf{C}_{\text{hole}}$ is the hole confidence that gives low confidence to pixels whose rays intersect empty space. We conducted an ablation experiment to compare the performance of our novel voxel-based confidence $\mathbf{C}_{\text{Vsize}}$ against the ambient occlusion confidence $\mathbf{C}_{\text{AO}}$ from [61]. We evaluated both approaches using different truncation thresholds on a small ScanNet++ [75] subset (*i.e.*, `07f5b601ee`, `08bbbdcc3d`, `0c5385e84b`, `210f741378`, `25aa952aa3`, `39f36da05b`, `56a0ec536c`, `5a269ba6fe`, `a1d9da703c`, `bc2fce1d81`, `be0ed6b33c`, `daffc70503`, `dc263dfbf0`, `ef18cf0708`, `fb564c935d`), reporting depth estimation results in Table 8. Notably, our $\mathbf{C}_{\text{Vsize}}$ consistently achieves lower MAE while maintaining a higher density if compared to $\mathbf{C}_{\text{AO}}$. We selected $\mu_{\text{Vsize}} = 0.75$ as the final truncation threshold for our voxel-based confidence.

Training Method	SE-CFF				E-FoundationStereo			
	1PE ↓	2PE ↓	3PE ↓	MAE ↓	1PE ↓	2PE ↓	3PE ↓	MAE ↓
MIX 3 (SVRaster [59])	**24.73**	**8.58**	**5.08**	**1.01**	**20.99**	**6.82**	**4.10**	**0.89**
MIX 3 (Depth Anything v3 [40])	74.35	47.82	30.63	3.18	71.37	41.01	23.99	2.54
EV-SceneFlow [17, 45]	66.30	50.18	41.47	3.50	61.80	48.04	41.68	3.10
EV-(SceneFlow+TartanAir) [17, 45, 65]	57.78	33.01	19.75	2.17	41.86	23.13	16.58	1.76

Table 9. **Further in-domain experimental results – DSEC dataset [18].** On top: comparison between SVRaster and Depth Anything v3 generated data. At the bottom: results by extending the synthetic data used to generate proxy events.

Model	Parameters (M)	FLOPs (G)	Runtime (ms)	Peak Memory (MB)
SE-CFF [49]	2.97	85.98	46.27	379.13
EMatch [82]	6.71	501.95	115.20	3090.49
E-StereoAnywhere	39.96	1566.58	219.81	1479.82
E-FoundationStereo	60.09	4445.51	280.11	1525.13

Table 10. **Hardware analysis on DSEC dataset [18].** Measurements taken on a X GPU.

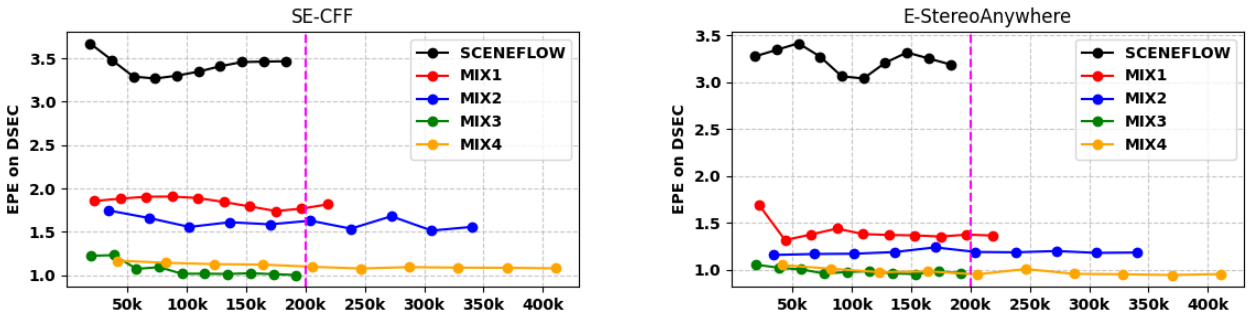


Figure 7. **Evaluation on DSEC after different numbers of training steps.**

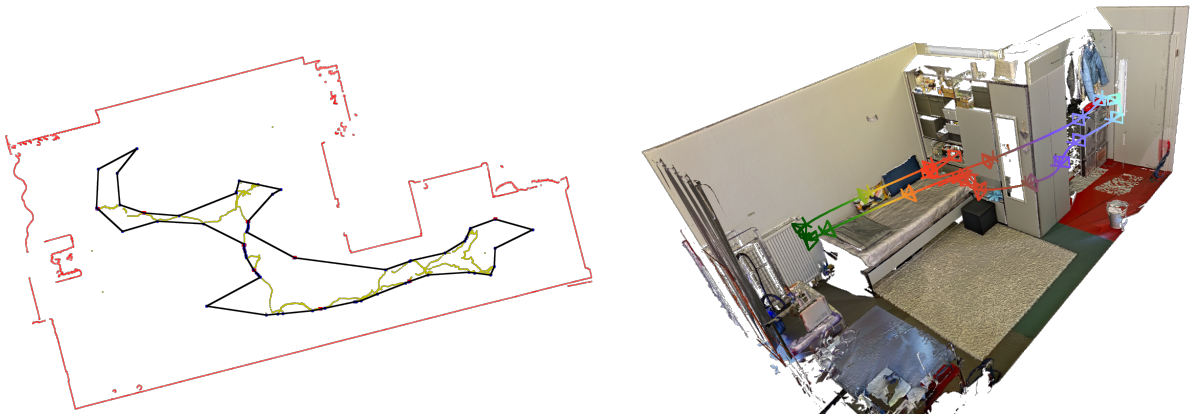
7. Additional Experiments

We now report further, focused experiments.

Further in-domain comparisons. Table 9 reports some additional experiments on DSEC, aimed at assessing the impact of rendering quality on the accuracy of the trained stereo models. We conduct this further evaluation over two axis: on top, we compare the results achieved by replacing SVRaster as the rendering engine of our pipeline with the feed-forward model Depth Anything v3 [40]. Despite the much faster data generation process enabled by this latter, we can observe a significant drop in the accuracy of the trained models; at the bottom, we extend the amount of synthetic data used to generate proxy events with E2VID [17], specifically by including TartanAir together with Sceneflow. Despite the improvement enabled by the larger amount of initial data, we can still notice a consistent gap between models trained on this kind of data with respect to ours. Importantly, we emphasize that event data generated from synthetic RGB datasets are not direct competitors to our EventHub framework; rather, the two sources could be combined to enhance performance further.

Efficiency Analysis. In Table 10, we report the complexity of each of the stereo backbones involved in our experiments, detailing the number of parameters, FLOPs, the runtime and the peak memory usage. SE-CFF stands as the least complex architectures, although achieving the worse results in our evaluation. On the contrary, E-StereoAnywhere and E-FoundationStereo stand as the most computationally intense architectures.

Convergence Analysis. By fixing the amount of epochs across the different dataset to 10, as described in the main paper, we obtain different amounts of total training steps, possibly biasing the evaluation of the trained models. However, as shown in Figure 7, we can appreciate how the models converge pretty soon to stable results, with marginal or no improvements being achieved by extending the training for more iterations, as occurs when using larger data splits such as MIX2 and MIX4.



2D top-view showing the α-shape (black) Final 3D global trajectory inside the scene

Figure 8. **Global trajectory construction.**

8. Additional Details Concerning Implementation and Experimental Settings

In this section, we include additional implementation details, in particular an extended overview of our global trajectory $\Omega(\tau)$, the datasets splits used for both training [18, 61, 75] and evaluation [7, 18, 84], and the adaptation of $\mathcal{L}_{\text{NS}}$ loss for each stereo architecture, *i.e.*, SE-CFF [49], EMatch [82], E-StereoAnywhere [6], and E-FoundationStereo [69].

8.1. Global Trajectory Implementation

For each selected ScanNet++ scene, we gather all COLMAP training poses $[\hat{\mathbf{R}}_i|\hat{\mathbf{t}}_i] = \hat{\mathbf{T}}_i \in \mathbb{SE}(3)$ and project $\hat{\mathbf{t}}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)^\top$ onto a 2D top-view by discarding the last $\hat{z}_i$ component. We then compute the corresponding α-shape, yielding an obstacle-avoiding 2D circular path. The resulting 2D curve is lifted back to 3D via a nearest-neighbor search, which we used to optimize the three splines using least squares (implemented using the SciPy package). As detailed in the main paper, one spline provides a continuous representation of the translation component $\mathbf{t}_\tau$, while the splines $\mathbf{r}(\tau)$ and $\mathbf{l}(\tau)$ parametrize the rotation component $\mathbf{R}_\tau$:

$$\mathbf{R}_\tau = \begin{bmatrix}\mathbf{d}(\tau) \times \mathbf{l}(\tau) & \mathbf{d}(\tau) & \mathbf{l}(\tau)\end{bmatrix}, \quad \mathbf{d}(\tau) = \mathbf{l}(\tau) \times \mathbf{r}(\tau). \tag{17}$$

However, given the ScanNet++ randomness of pose orientation, which causes unnatural camera egomotion, we re-estimate camera rotations $\hat{\mathbf{R}}_i$ to align them with the direction of motion. Specifically, we approximate the motion direction $\nabla\mathbf{t}_\tau$ using finite differences, and construct the updated orientation:

$$\mathbf{R}'_\tau = \begin{bmatrix}\mathbf{r}'(\tau) & \mathbf{d}'(\tau) & \nabla\mathbf{t}_\tau\end{bmatrix}, \quad \mathbf{r}'(\tau) = \mathbf{g} \times \nabla\mathbf{t}_\tau, \quad \mathbf{d}'(\tau) = \nabla\mathbf{t}_\tau \times \mathbf{r}'(\tau), \tag{18}$$

where $\mathbf{g} = (0\ 0\ 1)^\top$ denotes the ScanNet++ gravity vector. Finally, we clamp the $\hat{z}_i$ translation component to its $[45, 55]$-th percentile range to suppress strong vertical oscillations. This procedure yields a "human-like" walking trajectory through the scene, as shown in Figure 8 (right).

8.2. ScanNet++ Scenes Used for NVS

We enrich Section 4.1 with further information regarding the dataset used for event data generation [18, 61, 75]. For event data generation from Novel View Synthesis, we collect 30 samples for each scene, where each sample is composed of the stereo streams $\mathbf{E}_L$ and $\mathbf{E}_R$, the intrinsic $\mathbf{K}$, the baseline b, the RGB triplet $\mathbf{I}_{LL}$, $\mathbf{I}_L$, and $\mathbf{I}_R$, the depth $\mathbf{Z}$ and the confidence $\mathbf{C}_{\text{Vsize}}$. The maximum number of events for the event stereo streams $\mathbf{E}_L$ and $\mathbf{E}_R$ is limited to 650 000 and 1 000 000 events, respectively, for the samples at resolutions 640×480 px and 1280×720 px. Furthermore, we randomize the contrast threshold using a uniform distribution $\mathcal{U}(0.15, 0.25)$. We used all 270 scenes from the NeRF Stereo Dataset [61] – *i.e.*, starting from scene 0000 up to scene 0269 – while we selected the following 403 scenes from ScanNet++[75]: `00777c41d4`, `0271889ec0`, `02c2ddee2a`, `036bce3393`, `0452249a1e`, `04d0dc245b`, `04df8734b7`, `052d72e137`, `0658da5bc0`, `068ba2946c`, `06b5863f73`, `06bc6d1b24`, `076c822ecc`, `079a326597`, `07f5b601ee`, `08bbbdcc3d`, `09a6767fc2`, `09bced689e`, `0a5c013435`, `0c5385e84b`, `0c6c7145ba`, `0c7962bd64`, `0caa1ae59a`, `0d8ead0038`, `0e100756bf`, `0e350246d3`, `0e900bcc5c`, `0f0191b10b`,

0f25f24a4f, 0f3474b837, 10242d1eaf, 10c8ab99f4, 1117299565, 1204e08f17, 124a6e789b, 12c0f7a7da, 13285009a4, 132cb783ed, 13b4efaf62, 15c4aa5bbb, 16c9bd2e1e, 1730c7d709, 1841a0b525, 192ab15daf, 1a130d092a, 1a3100752b, 1a8e0d78c0, 1b9692f0c7, 1bb93d185e, 1c08823a41, 1c4b893630, 1c7a683c92, 1d003b07bd, 1eacc65607, 20871b98f3, 20ff72df6e, 210f741378, 216b9e55e8, 238b940049, 246fe09e98, 2489b7f4fe, 24b248e676, 251443268c, 25aa952aa3, 25bae29ab3, 25bde9e167, 260db9cf5a, 260fa55d50, 2634683a9f, 2748de13fb, 2779f8f9e2, 27dc178a3d, 281bc17764, 2970e95b65, 29c7afafed, 2a1b555966, 2a496183e1, 2b71155e0d, 2f5996ff01, 2f6f83ea1f, 302a7f6b67, 303745abc7, 30f4a2b44d, 320c3af000, 324d07a5b3, 3391ff8a71, 3423e509af, 35050f41c5, 355e5e32db, 364f01bc18, 37562e7f48, 3799bd47b3, 37c9538a2b, 38fcf02d0b, 390eda9157, 39580e2a43, 39e6ee46df, 39f36da05b, 3a3745a437, 3aa115e55e, 3b90310b1c, 3c8d535d49, 3caf4324fd, 3cbb18c391, 3ce6d36ab5, 3d838ee1ab, 3e7e4b07c4, 3e928dc2f6, 3ff873c77e, 413085a827, 41b00feddb, 4318f8bb3c, 4380e4646a, 43cd995c51, 4422722c49, 4423a61d09, 442b144761, 44c85584ae, 4517d988d8, 45d2e33be1, 46001f434d, 4610b2104c, 46638cfd0f, 47b37eb6f9, 4808c4a397, 480ddaadc0, 484ad681df, 48573f4c95, 48701abb21, 4897e95232, 49789448b8, 4aef651da7, 4c141d5b1b, 4c5c60fa76, 4d451d9c36, 4e0b8cbd33, 4ea827f5a1, 504cf57907, 511061232, 51bdbf173f, 523657b4d0, 5334a4164a, 53755e535e, 546292a9db, 54b005d19d, 55b2bf8036, 5654092cc2, 56669a70bc, 56a0ec536c, 58960ff105, 589f5c7c58, 58f6a5c5ec, 59e3f1ea37, 5a269ba6fe, 5a9cdde1ba, 5aeac3800a, 5bc6227191, 5c215ef3b0, 5d152fab1b, 5d902f1593, 5ea3e738c3, 5f0fb991a7, 6126572846, 612f70fe00, 617326da3e, 618310ed87, 6183f0657d, 61adeff7d5, 6248c6742d, 635852d56e, 639f2c4d5a, 6464461276, 64672b5bf5, 652d9cb0d7, 666d04a14a, 66ba53719a, 66c98f4a9b, 67d702f2e8, 696317583f, 69e56cf0f8, 69e5939669, 6ad6cef000, 6b19334aeb, 6b40d1a939, 6bd39ac392, 6da1d5ab04, 6f1848d1e3, 70945f435a, 709ab5bffe, 70f0e494b2, 712b9ae775, 724c40236c, 72f527a47c, 73f9370962, 75d29d69b8, 7739004a45, 77b40ce601, 785e7504b9, 791a5c253d, 7b04052ad0, 7b4a316aea, 7b4cb756d4, 7c0ba828a9, 7c31a42404, 7c31bccde5, 7d8d37ca38, 7e7d2e8640, 7f22d5ef1b, 7f68c514bd, 7f77abce34, 7fb8ff20e9, 8013901416, 80ffca8a48, 81a82c3618, 82f448db76, 82ff39b7ef, 85251de7d1, 85dc2702b7, 867d97cf3d, 871efc90fa, 8737a0d1ad, 88627b561e, 8890d0a267, 88f265fe25, 893fb90e89, 8be0cd3817, 8d0f714398, 8de35c04a3, 8e22c48c20, 8f82c394d6, 8fc40ba77b, 9084d4cd97, 909a9ea5fc, 91fc568d84, 9444b90aaa, 9471b8d485, 94b1acde81, 95748dd597, 95d525fbfd, 97e5512e91, 9816c49e97, 98b4ec142f, 98fe276aa8, 99010a8938, 9b74afd2d2, 9bfbc75700, 9c7b4394af, 9cfea269dd, 9d8fcc4215, 9dc5ad040f, 9ef5fc6271, a08d9a2476, a1d9da703c, a23f391ba9, a30646cae6, a31b2ef388, a492fe77aa, a4d48ea6b3, a4e227f506, a892730b61, a8f7f66985, a9e4791c7e, aa852f7871, aab83fd6f1, ab046f8faf, ab11145646, ab6983ae6c, abf29d2474, ac250f0ead, acd69a1746, ad2d07fd11, adf4ab4a53, aea84db0de, b068706ef0, b08a908f0f, b09431c547, b0b004c40f, b0f057c684, b0fe0c610f, b1d75ecd55, b20a261fdf, b24697b3a1, b2632b738a, b3ac0beef0, b4b39438f0, b5918e4637, b6d73041c8, b97261909e, bac7ee3b1b, bb05a0c48c, bb0ad8a081, bc2fce1d81, bc400d86e1, be05b26a38, be0ed6b33c, be8367fcbe, bf07750a0b, bf50f418ba, bfcfe53c6a, bfd3fd54d2, c026d108e0, c07c707449, c08d1d52b7, c0da8f4a4d, c0f5742640, c29b5e479c, c2d714d386, c31ebd4b22, c40466a844, c465f388d1, c47168fab2, c4aaedcfd1, c4d4cb61f6, c601466b77, c842edbdf5, c856c41c99, c8eeef6427, c8f2218ee2, c9a8357e8f, ca0c580422, cab239278a, cb7785f6ad, cc5ea8026c, ccfd3ed9c7, cd0b6082d2, ce12db9e81, cec8312f4e, d054227009, d1345a65c1, d1f82299d0, d240136ce4, d2f44bf242, d537ef1d41, d551dac194, d61691f945, d6a77f7c22, d6bb698875, d7abfc4b17, d7b871aaa8, d807fb583b, d918af9c5f, d986399f4c, daffc70503, db5293a870, dc263dfbf0, dd685be466, de3c77cecd, de5881aa12, deb1867829, dec0b11090, defd3457db, dfa70fb232, dfac5b38df, e050c15a8d, e0de253456, e1aa584dd5, e2caaaf5b5, e3ad7115db, e3b3b0d0c7, e3c1da58dd, e3e0617f98, e3ecd49e2b, e3ef8b690b, e4007ff6b5, e4e625a3e4, e4fb2a623b, e5a769dbf5, e667e09fe6, e69064f2f3, e7ccd75e5d, e81c8b3eec, e8e81396b6, e8ea9b4da8, e909f8213d, e9e16b6043, eaa6c90310, eaab7bcc15, eab5494dca, eb8ef9b4cc, ec2cb8dae1, ed2216380b, eea4ad9c04, eeeb9836b8, ef18cf0708, ef25276c25, f19ca0a52e, f248c2bcdc, f25f5e6f63, f2e6c43543, f38b0108a1, f3f016ba3f, f576071590, f6659a3107, f6a9b64a0d, f847086d15, f8d5147d1d, f8e13ab4ae, f8eac0ad24, f97de2c3e9, faba6e97d7, faec2f0468, fb152519ad, fb564c935d, fb893ffaf3, fb9b4c2f15, fd361ab85f, fd8560cfd6, fe1733741f, and ff17657f71.

8.3. Custom Stereo Losses

As mentioned in Section 6.2, we adapt $\mathcal{L}_{\text{NS}}$ for each event stereo model – *i.e.*, SE-CFF [49], EMatch [82], E-StereoAnywhere [6], and E-FoundationStereo [69]. In particular, we started from the original loss proposed by the authors of each architecture, obtaining the following losses:

- We adapt $\mathcal{L}_{\text{NS}}$ for SE-CFF [49] starting from their multi-scale disparity loss:

$$\mathcal{L}'_{\text{NS}} = \sum_{s}^{L} w_s \left[\left(\lambda_{\text{disp}} \cdot \eta(\mathbf{C}^{(s)}_{\text{Vsize}}; \mu_{\text{Vsize}}) \cdot \mathcal{L}^{(s)}_{\text{disp}} + \mathbf{M}^{(s)}_{\text{auto}} \cdot \lambda_{\text{3p}} \cdot (1 - \eta(\mathbf{C}^{(s)}_{\text{Vsize}}; \mu_{\text{Vsize}})) \cdot \mathcal{L}^{(s)}_{\text{3p}} \right) + \lambda_{\text{smooth}} \cdot \mathcal{L}^{(s)}_{\text{smooth}} \right], \quad (19)$$

where L is the number of scales, w_s is the weight for the s-th scale, $\mathcal{L}^{(s)}_{\text{disp}}$ is a L1 loss computed at scale s, $\mathcal{L}^{(s)}_{\text{smooth}}$ is a

gradient regularization term that ensure smooth disparity estimations, and $\lambda_{\text{smooth}} = 0.1$ is the weighting term for $\mathcal{L}^{(s)}_{\text{smooth}}$.

- For other stereo networks – *i.e.*, EMatch [82], E-StereoAnywhere [6], and E-FoundationStereo [69] – we adopt a RAFTStereo-like [42] loss with further supervision for the initial disparity estimation:

$$\mathcal{L}''_{\text{NS}} = \left[\sum_{i}^{N} w_i \left[\left(\lambda_{\text{disp}} \cdot \eta(\mathbf{C}_{\text{Vsize}}; \mu_{\text{Vsize}}) \cdot \mathcal{L}^{i}_{\text{disp}} + \mathbf{M}_{\text{auto}} \cdot \lambda_{\text{3p}} \cdot (1 - \eta(\mathbf{C}_{\text{Vsize}}; \mu_{\text{Vsize}})) \cdot \mathcal{L}^{i}_{\text{3p}} \right) \right] \right] + \mathcal{L}^{0}_{\text{NS}}, \tag{20}$$

where N is the number of refinement steps, w_i is the exponentially increasing weight for the i-th refined disparity, $\mathcal{L}^{i}_{\text{disp}}$ is a L1 loss computed with respect to the i-th refined disparity, and $\mathcal{L}^{0}_{\text{NS}}$ is the NeRF-supervised loss for the initial disparity.

As mentioned in the main paper (Sec. 4.1), the losses $\mathcal{L}'_{\text{NS}}$ and $\mathcal{L}''_{\text{NS}}$ are used for NVS data only – where $\mathbf{C}_{\text{Vsze}}$, and the RGB triplet $\mathbf{I}_{LL}$, $\mathbf{I}_{L}$, and $\mathbf{I}_{R}$ are available. For the other sources of data – *i.e.*, distilled data from [18], and ground-truth supervised trainings – we maintain only the $\mathcal{L}^{(s)}_{\text{disp}}$ and $\mathcal{L}^{i}_{\text{disp}}$ terms respectively from $\mathcal{L}'_{\text{NS}}$ and $\mathcal{L}''_{\text{NS}}$.

9. Additional Qualitative Results

In this section, we collect additional qualitative results, including full samples from the EventHub data (Sec. 9.1), predictions generated by event-based stereo networks (Sec. 9.2), and finally, a qualitative comparison between conventional RGB Stereo Foundation Models like FoundationStereo before and after fine-tuning on EventHub data against challenging night sequences (Sec. 9.3).

9.1. Qualitative Samples from EventHub

We report a few training samples generated with our EventHub pipeline, obtained both by means of cross-modal distillation and by deploying novel view synthesis.

Figure 9 shows three samples from the DSEC datasets, obtained through the former paradigm. From left to right, we display the left image from the color stereo pair, the left event frame, and the proxy disparity map generated by Foundation-Stereo [69] and projected over the event frame, as described in Sec. 3.1.2. We can notice, in particular, the high level of detail of these predicted labels, crucial for providing the event stereo models with strong guidance.

Figures 10 and 11 collect four examples from scenes available in the NeRFStereo [61] and ScanNet++ [75] datasets, respectively. From left to right, we show rendered RGB and event frames, followed by rendered depth maps, confidence maps based on voxel sizes, and rendered depth maps masked according to confidence thresholding. The latter further highlight the importance of confidence thresholding in removing outliers in the rendered depth maps.

9.2. Predictions from Event Stereo Networks

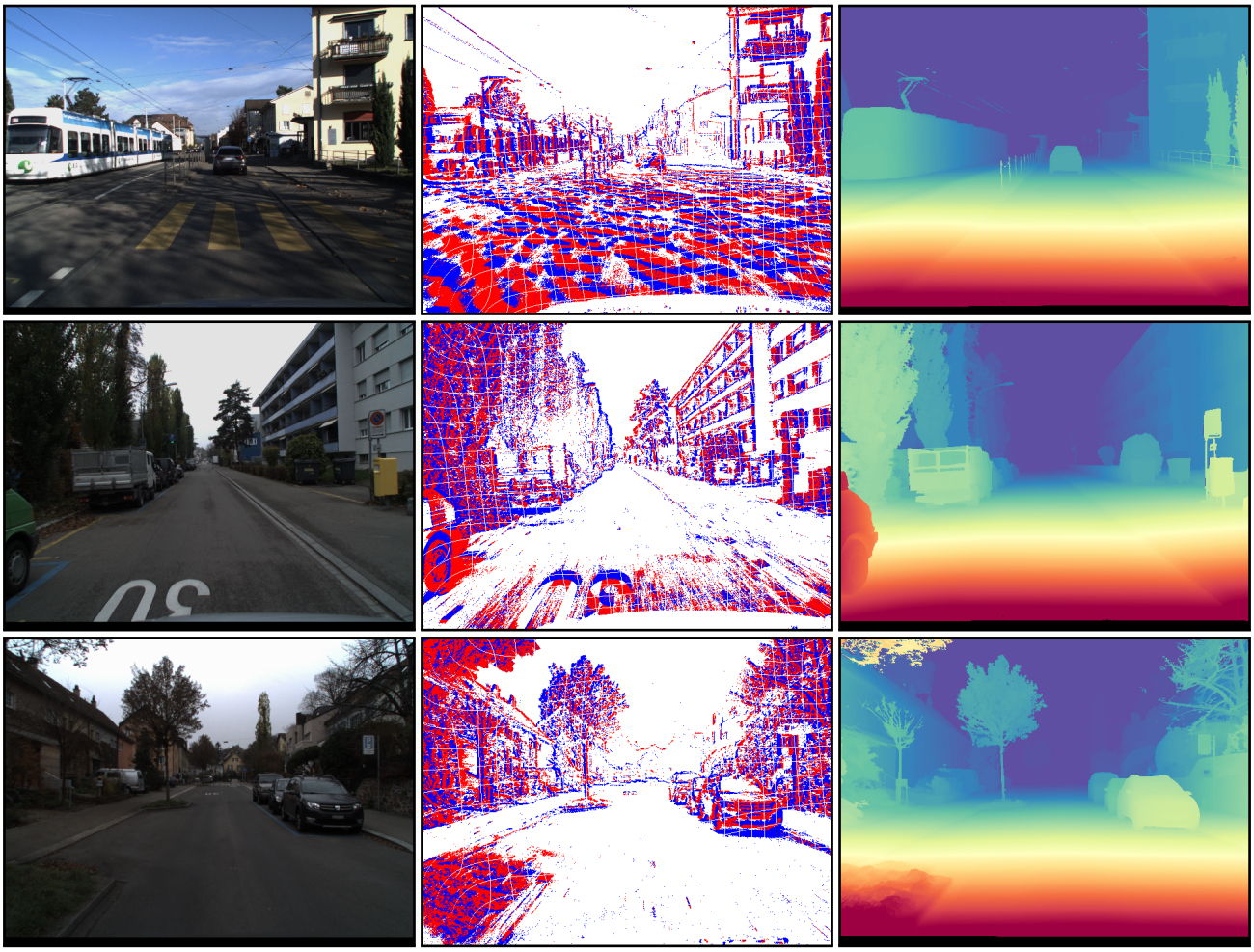
We report additional qualitative results concerning event stereo models trained with different supervision flavors.

Figures 12 to 14 collect two samples each, respectively, from DSEC [18], M3ED [7] and MVSEC [84] datasets. On any dataset, we can clearly notice how MIX 4 allows for training any of the four models involved in our experiments at their best, with the novel models introduced by repurposing stereo foundation models from the RGB literature [6, 69] – E-StereoAnywhere and E-FoundationStereo – benefiting the most from the superior annotations produced by EventHub.

9.3. Predictions from RGB SFMs at Night

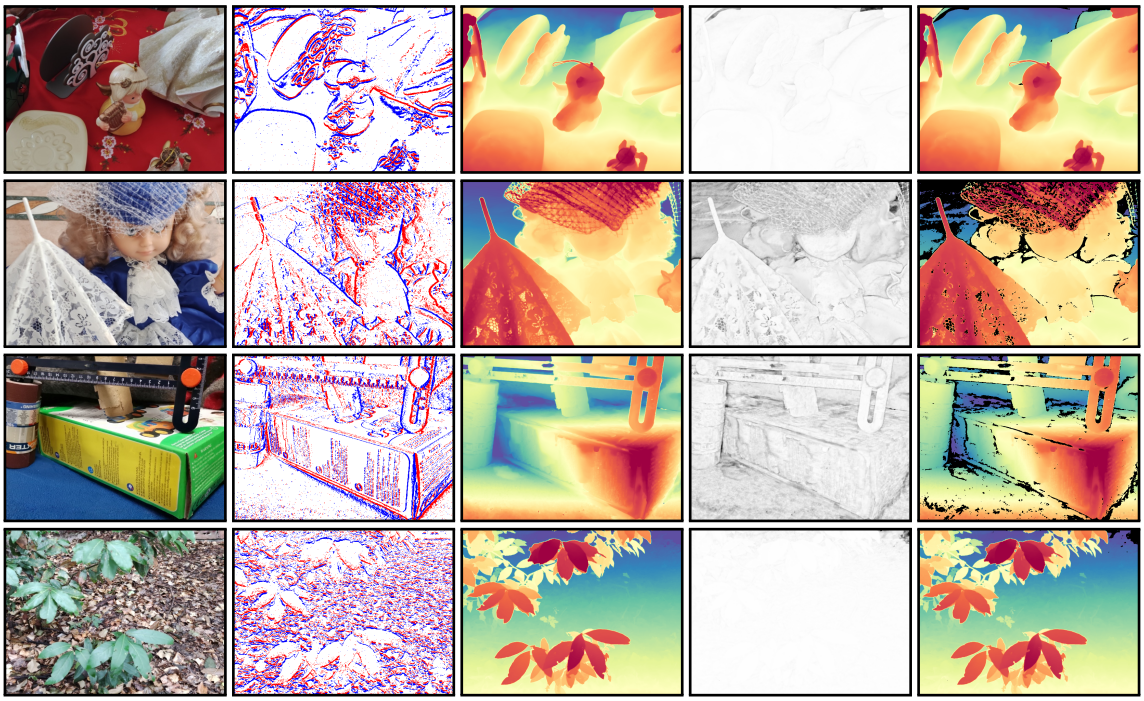
We conclude by showing qualitatively how we can improve the original stereo foundation models – StereoAnywhere and FoundationStereo, from which we derived our new E-StereoAnywhere and E-FoundationStereo frameworks – on challenging conditions where they struggle, by distilling the knowledge of E-StereoAnywhere and E-FoundationStereo themselves.

Figures 15 and 16 collect two nighttime images from DSEC [18] each. From left to right, we show (a) the left color image, then the predictions by FoundationStereo [69] respectively (b) before any further fine-tuning – i.e., using the original weights – and (c) after being fine-tuned on proxy labels distilled by E-FoundationStereo. After fine-tuning, FoundationStereo learns to deal with this challenging domain and is able to better retain fine details in the predicted disparity maps.



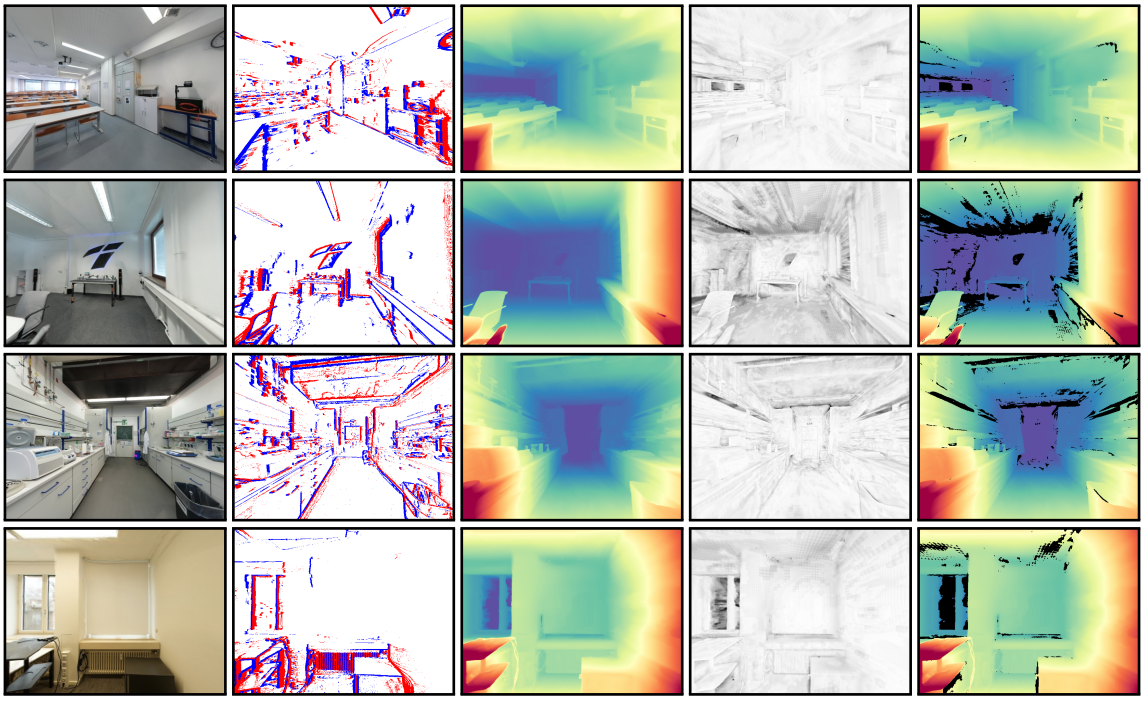
(a) (b) (c)

Figure 9. **Qualitative examples of training data generated by EventHub on DSEC [18].** (a) left color image, (b) left event frame, (c) proxy disparity labels predicted by FoundationStereo [69] and projected on the event frame.



(a) (b) (c) (d) (e)

Figure 10. **Qualitative examples of training data generated by EventHub on NeRFStereo [61].** (a) rendered color image, (b) rendered event frame, (c) rendered depth and (d) confidence (the brighter, the higher the confidence in the estimated depth values), and (e) rendered depth masked according to confidence thresholding.



(a) (b) (c) (d) (e)

Figure 11. **Qualitative examples of training data generated by EventHub on ScanNet++ [75].** (a) rendered color image, (b) rendered event frame, (c) rendered depth and (d) confidence, (e) rendered depth masked according to confidence thresholding.

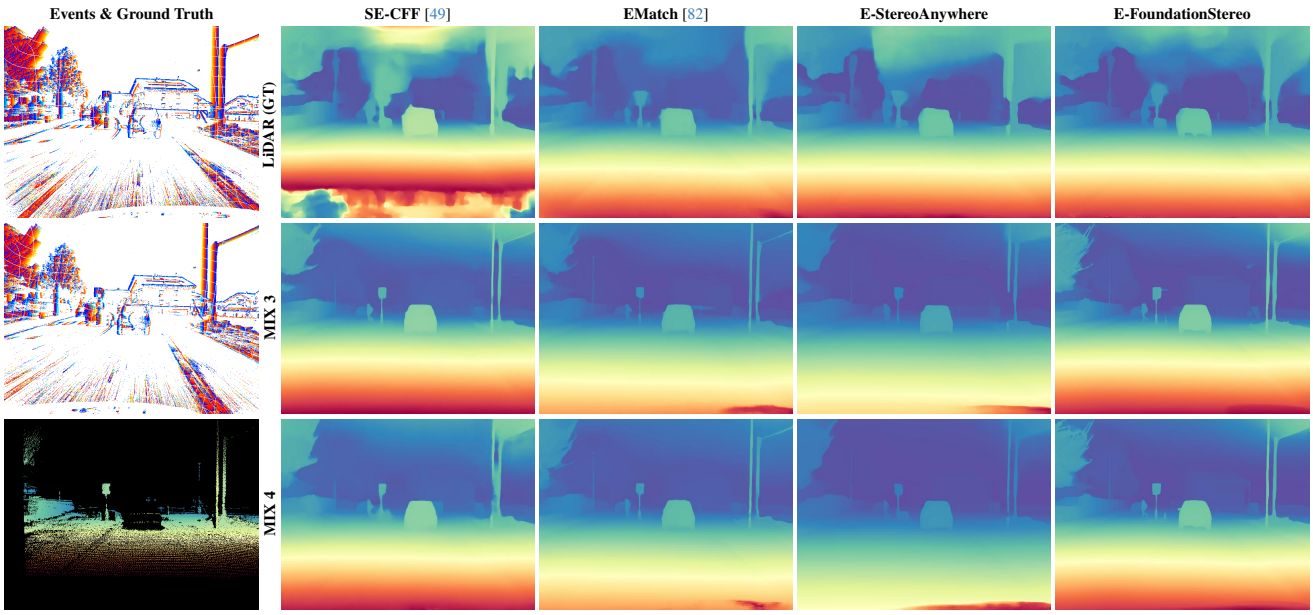


Figure 12. **Qualitative results on DSEC [18] dataset.** Predictions by the four models trained with LiDAR labels, MIX 3 or MIX 4.

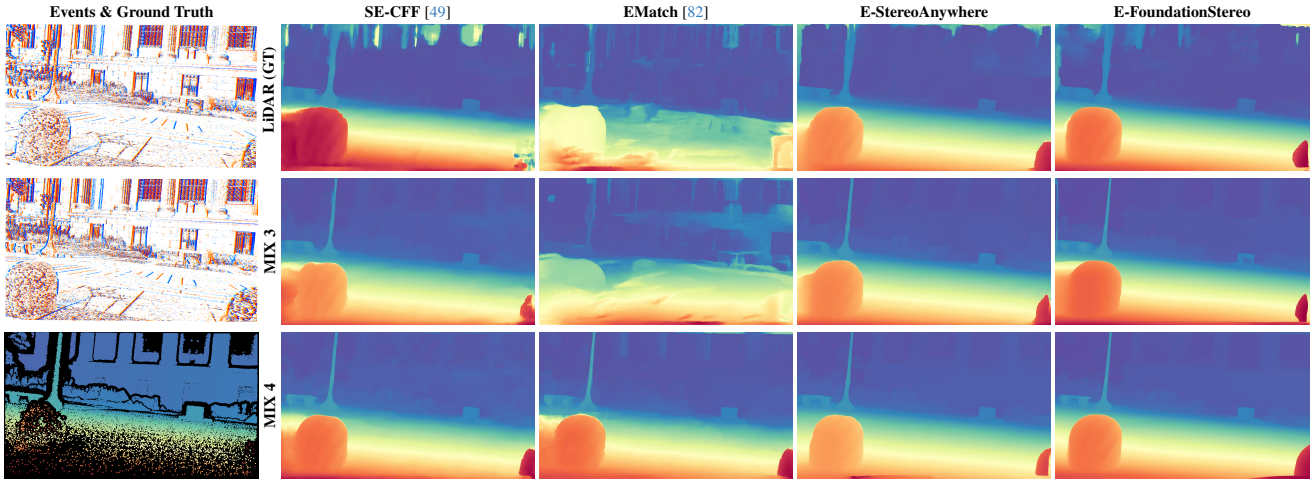


Figure 13. **Qualitative results on M3ED [7] dataset.** Predictions by the four models trained with LiDAR labels, MIX 3 or MIX 4.

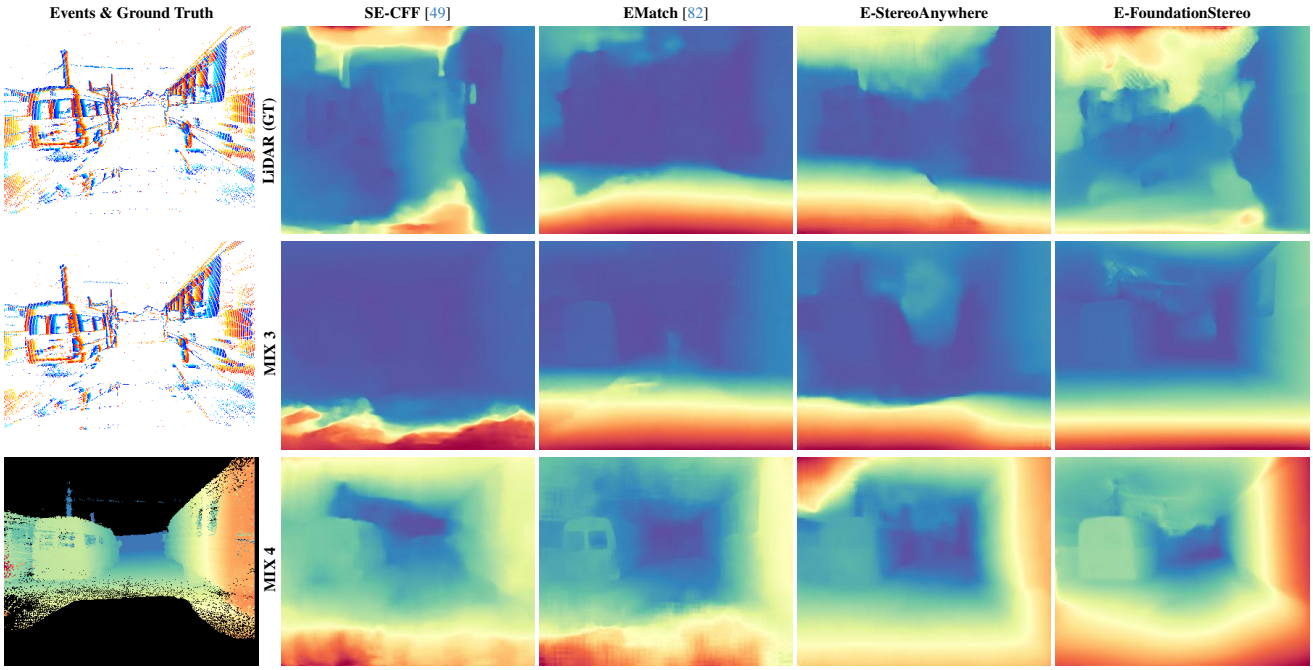
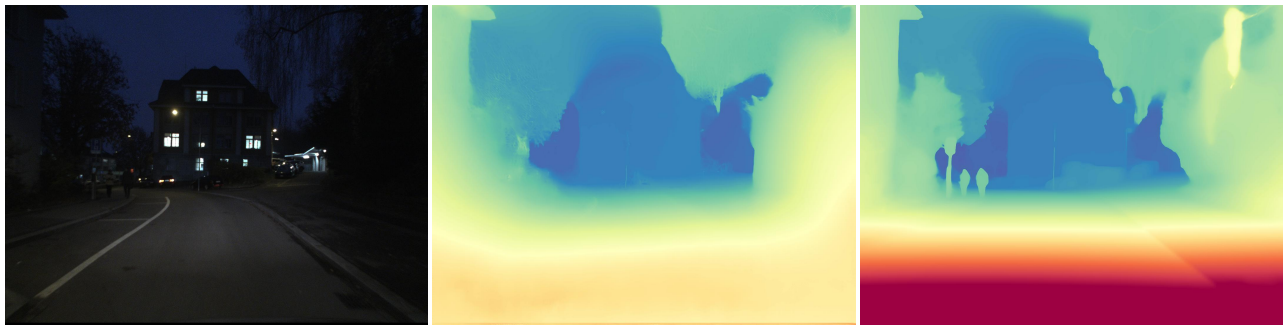


Figure 14. **Qualitative results on MVSEC [84] dataset.** Predictions by the four models trained with LiDAR labels, MIX 3 or MIX 4.

(a) (b) (c)

Figure 15. **Improved RGB FoundationStereo at Night.** Qualitative comparison on the `zurich_city_09_d` night sequence. (a) left RGB, (b) prediction by baseline FoundationStereo VIT-L, and (c) its fine-tuned counterpart using proxy labels from E-FoundationStereo VIT-S.



(a) (b) (c)

Figure 16. **Improved RGB FoundationStereo at Night.** Qualitative comparison on the `zurich_city_10_b` night sequence. (a) left RGB, (b) prediction by baseline FoundationStereo VIT-S, and (c) its fine-tuned counterpart using proxy labels from E-FoundationStereo VIT-S.