DELTA: Dense Depth from Events and LiDAR using Transformer's Attention

Supplementary Material

6. Enlarged Views of the Results on SLED, MVSEC, and M3ED

As described in the main article, an enlarged version of Fig. 3 is given in Fig. 7 (this version also includes the input LiDAR and event data), an enlarged version of Fig. 4 is given in Fig. 8, and an enlarged version of Fig. 5 is given in Fig. 9.

7. Alternative Versions of DELTA

Alternative versions of our DELTA network are given in Figs. 10 to 15. Versions of the network illustrated in Figs. 10 to 14 are used as part of the ablation study in Sec. 4.6 of the main article, while the version illustrated in Fig. 15 is used as part of Sec. 8 of this Supplementary Material.

8. Ablation Study on Encoding Heads

In addition to the ablation studies conducted in the main article, we propose here an additional variant of the network, DELTA$^{\text{NEH}}$, showcased in Fig. 15. Here, the convolutional encoding heads are replaced by a more direct splitting into patches, as originally done in the Vision Transformer [12]. To compensate for the reduced number of parameters in the network, we add a third layer of self-attention modules. At the end of the decoding, the patches are simply grouped back to an image-based format, and a final small convolutional head reduces the number of channels and smooths the resulting depth maps.

Results of DELTA$^{\text{NEH}}$ on the SLED dataset are shown in Tab. 7. As can be seen, DELTA$_{\text{SL}}^{\text{NEH}}$ does not perform well at all, even worse than all the variants showcased in the main article. As visually illustrated in Fig. 16, DELTA$_{\text{SL}}^{\text{NEH}}$ produces depth maps with large errors (especially for the tunnel in the bottom row), and where the junction between the patches remains visible, creating numerous artifacts. While a simple splitting into patches can be conducted for the Vision Transformer, as classification is the end task, here we require a dense reconstruction at the end, *i.e.*, we need to keep information about the structure of the scene. Therefore, in our case, computing the patches using a convolutional head allows for a better re-grouping of the patches at the end of the network, by allowing the decoding head to be guided by the corresponding data from the encoding head through our use of the convex upsampling module of [42].

9. Computational Complexity

We report in Tab. 8 several metrics of the computational complexity of DELTA, computed on a single NVIDIA L40 GPU. For high- (1280×720), mid- (640×480), and low-resolution (346×260) data, DELTA has a mean inference rate of 6.3Hz, 20.5Hz, and 47.8Hz respectively. Compared to the method of Cui *et al*. [10] with its reported output rate of 56Hz on the MVSEC dataset, our method is only 1.17 times slower, but for a much better accuracy as shown in Tab. 3 of the main article. Compared to ALED, despite the significant increase in the number of parameters due to the use of attention modules, DELTA requires a similar amount of FLOPS and of GPU memory, allowing its deployment on standard consumer-grade GPUs. Inference times of ALED are of course smaller, and while we can not exactly call our method real-time, we want to remind the reader here that the focus of our work was set on accuracy rather on real-time compatibility. As such, inference time and/or memory usage could be further reduced, as we are not using advanced optimizations like `torch.compile()`, and as we believe the DELTA architecture could be slightly revised to reduce its number of parameters while keeping a similar accuracy. Implementation on specialized hardware could also be considered for real-time robotic applications, but is beyond the scope of this work.

Map	Cutoff	DELTA$_{\text{SL}}$	DELTA$_{\text{SL}}^{\text{NEH}}$
Town01	10m	**0.66**	1.58 (+0.92)
	20m	**1.33**	2.89 (+1.56)
	30m	**1.91**	3.90 (+1.99)
	100m	**3.22**	6.73 (+3.51)
	200m	**4.54**	9.85 (+5.41)
Town03	10m	**0.54**	2.13 (+1.59)
	20m	**1.31**	3.17 (+1.86)
	30m	**1.93**	3.89 (+1.96)
	100m	**3.40**	6.14 (+2.74)
	200m	**4.63**	9.23 (+4.60)

Table 7. Absolute and relative mean depth errors (in meters) on the SLED dataset, for the base version of DELTA and the "No Encoding Head" variant shown in Fig. 15.

10. Additional Visual Results on the SLED Dataset

Additional qualitative results on the SLED dataset are given in Figs. 17 to 20. We showcase in Figs. 17 and 18 scenes with accurate estimations, but also some small and larger failure cases in Figs. 19 and 20.

Model	Resolution (with padding)	Dataset(s)	Patch size	Nb. param.	FLOPS	Inference time	Max. GPU mem.
DELTA	1280 × 720	SLED, M3ED	16	180.9M	1786.1B	157.9ms ± 2.0ms	10.64 GB
	640 × 480	DSEC	16	180.9M	596.4B	48.7ms ± 0.4ms	4.68 GB
	346 × 260 (348 × 264)	MVSEC	12	181.2M	300.4B	20.9ms ± 0.2ms	3.09 GB
ALED	1280 × 720	SLED, M3ED	/	26.2M	1546.0B	91.2ms ± 16.8ms	7.02 GB
	640 × 480	DSEC	/	26.2M	515.3B	26.9ms ± 5.0ms	2.74 GB
	346 × 260 (352 × 264)	MVSEC	/	26.2M	155.9B	7.5ms ± 1.4ms	1.36 GB

Table 8. Computational complexity metrics (number of parameters, FLOPS, mean inference time, maximum GPU memory usage) for DELTA and ALED, for both high-, mid-, and low-resolution data.

11. Additional Visual Results on the MVSEC Dataset

Additional qualitative results on the MVSEC dataset are given in Fig. 21, showing the quality of the results for both day and night scenes despite the sparse and low-resolution input event and LiDAR data.

12. Additional Visual Results on the M3ED Dataset

Additional qualitative results on the M3ED dataset are given in Figs. 22 and 23, where the sparsity of the ground truth depth maps (especially compared to the density of the LiDAR data) and the blob-like appearance of the objects in the predictions can be better observed.

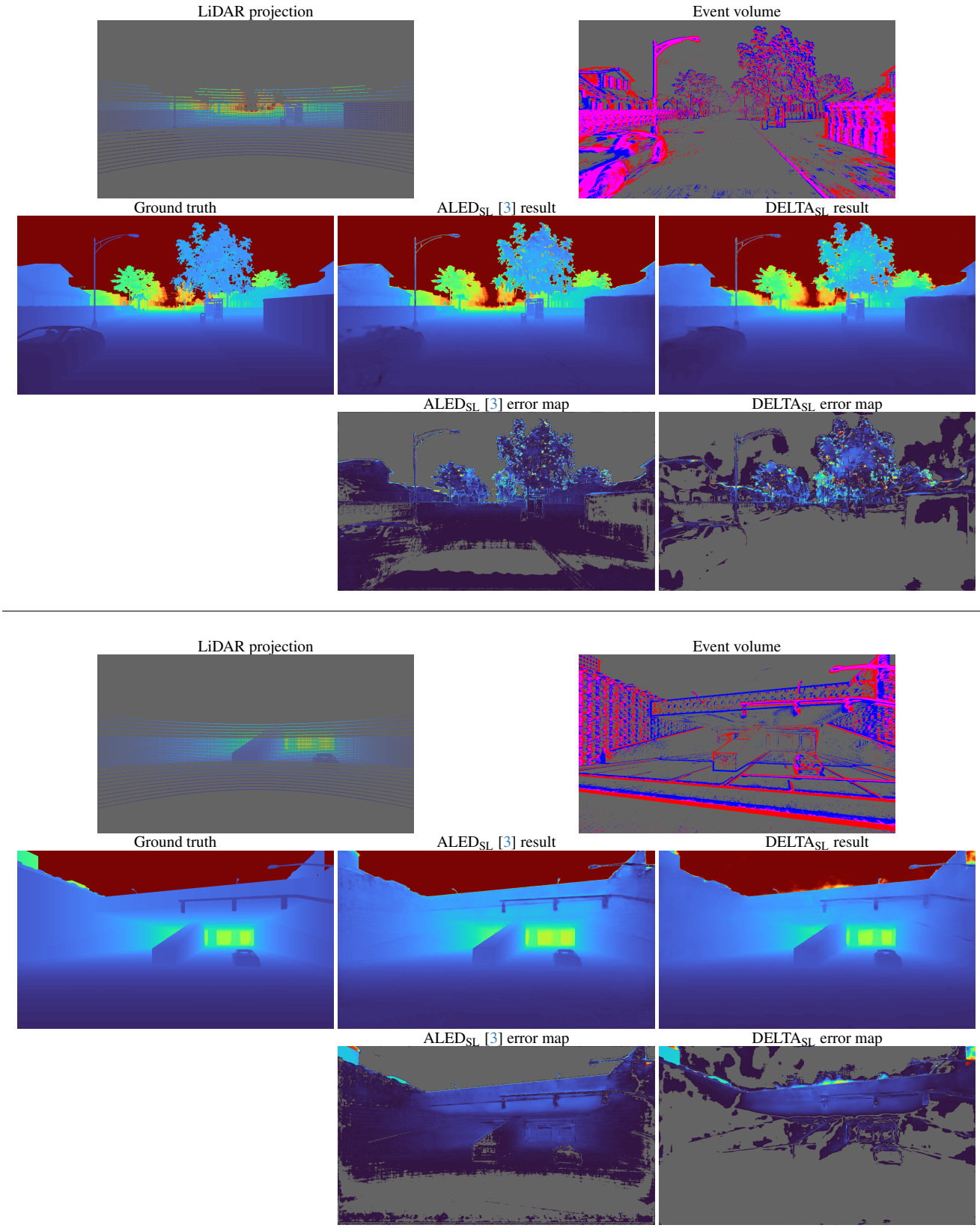

Figure 7. Comparison on the `Town01_08` (top) and `Town03_19` (bottom) sequences of SLED (enlarged version of Fig. 3).

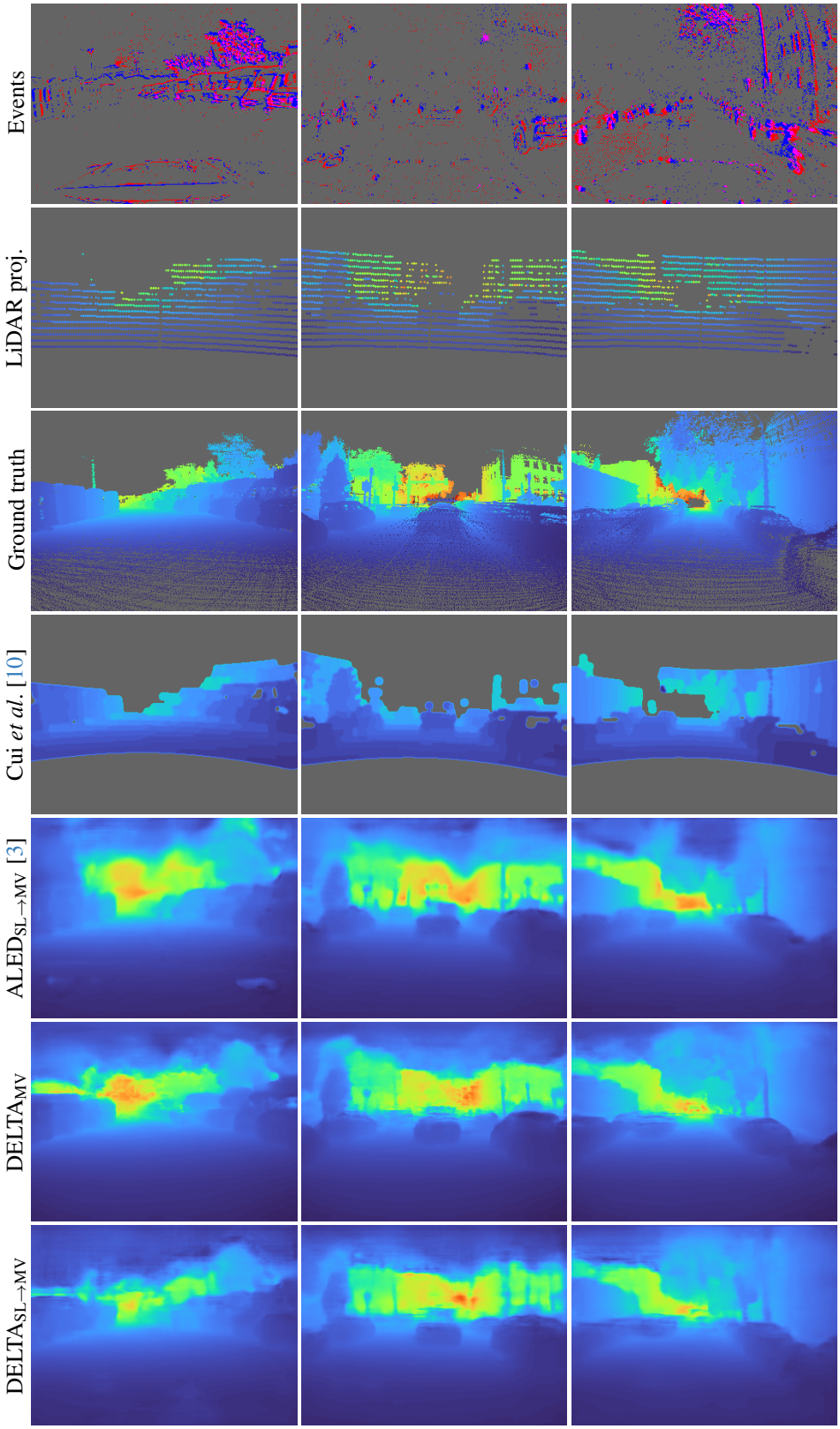

Figure 8. Comparison on the `outdoor_day_1`, `outdoor_night_1`, and `outdoor_night_2` sequences of MVSEC (enlarged version of Fig. 4).

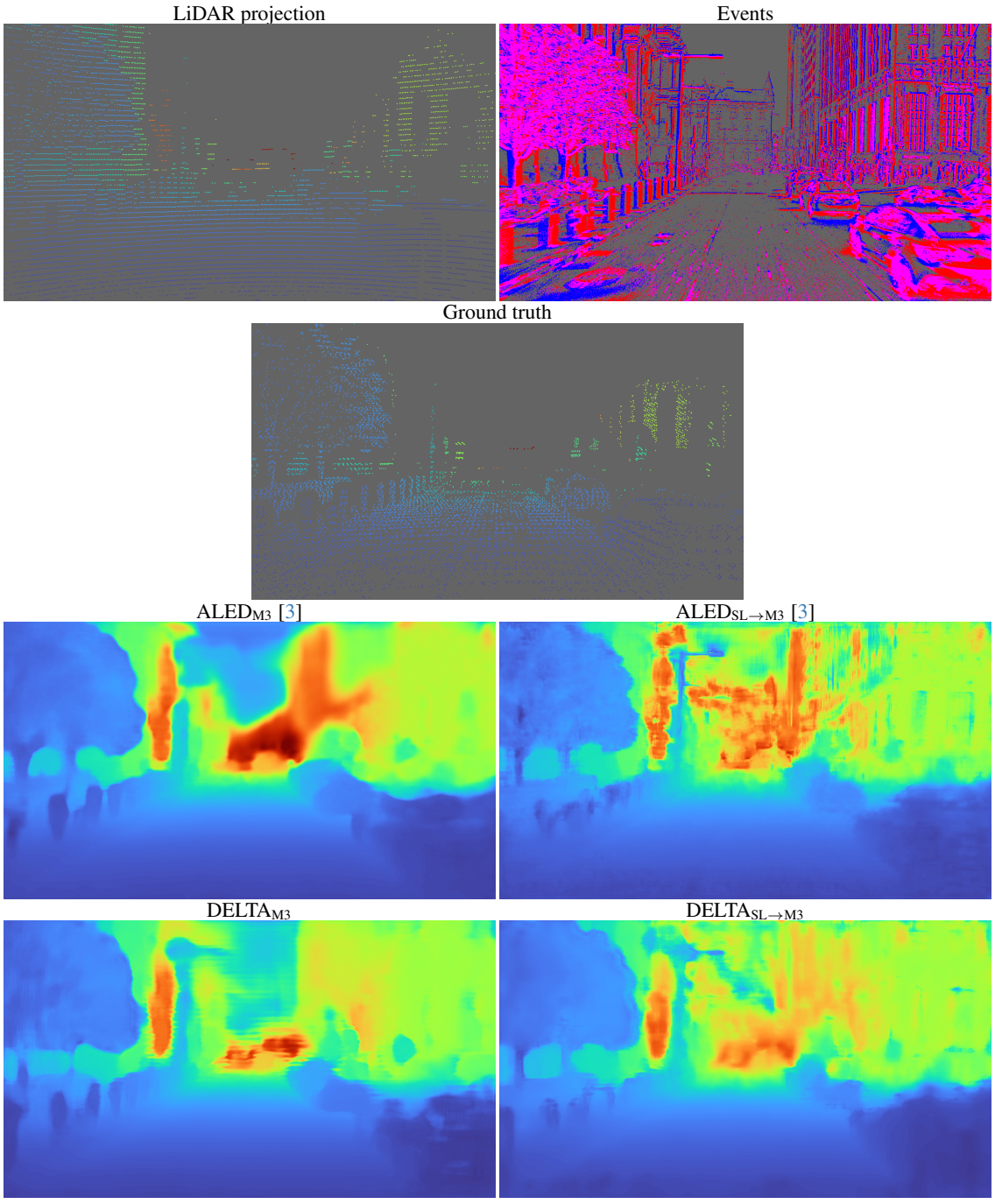

Figure 9. Comparison on the `city_hall_day` sequence of M3ED (enlarged version of Fig. 5).

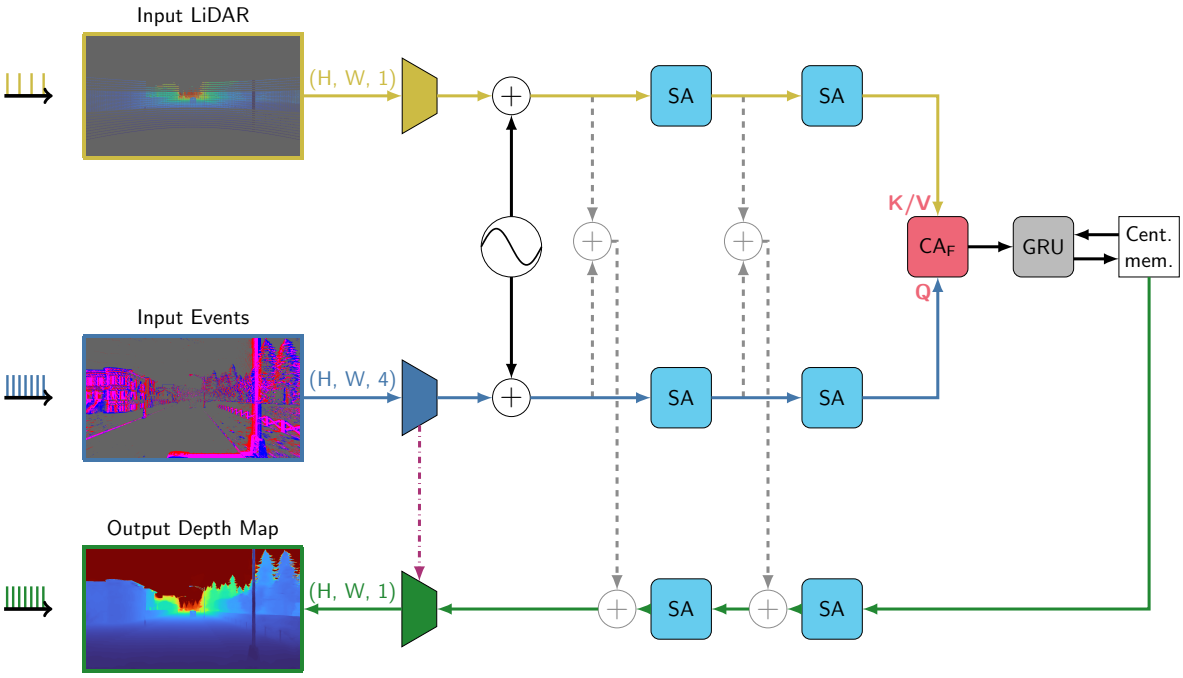

Figure 10. The alternative architecture without propagation memory, DELTA^{NPM}.

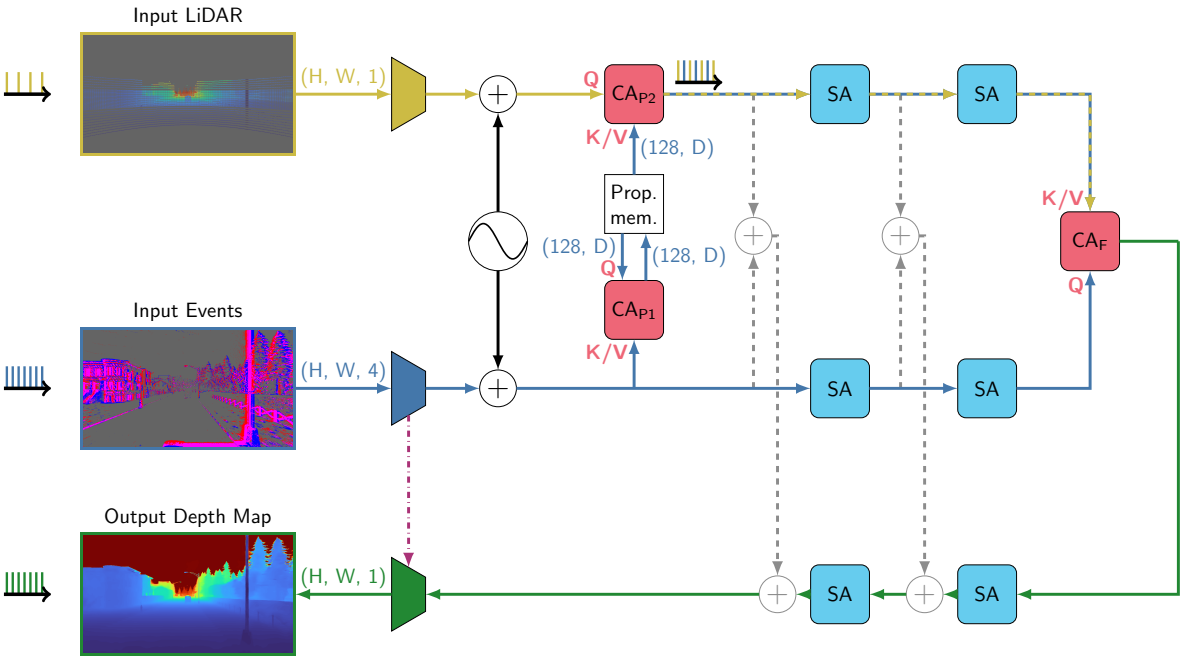

Figure 11. The alternative architecture without central memory, DELTA^{NCM}.

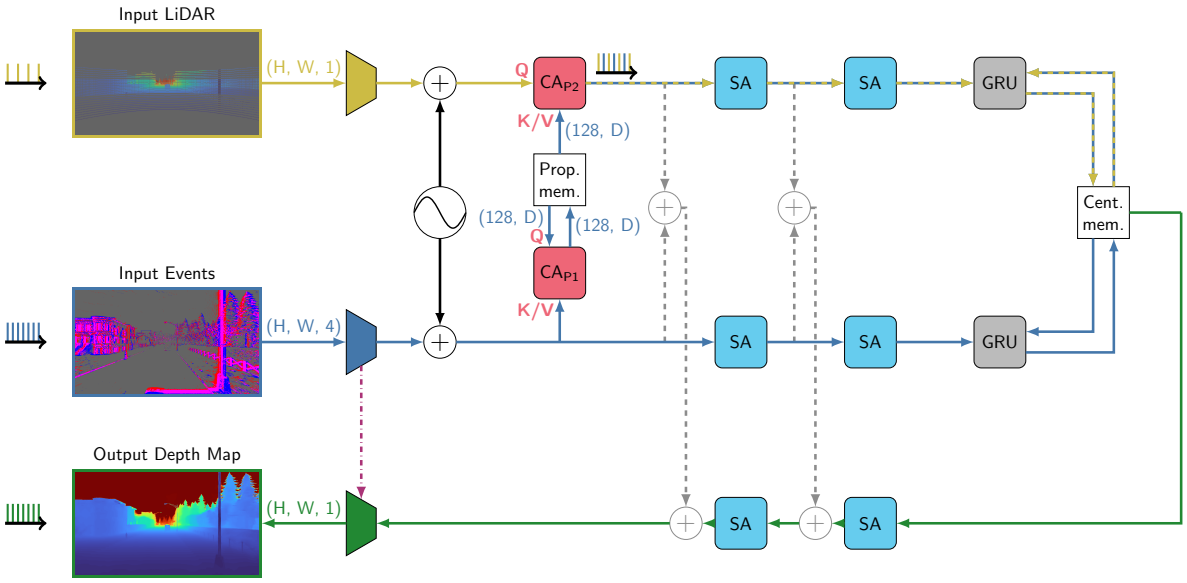

Figure 12. The alternative architecture without the central cross-attention, DELTA^{NCA}.

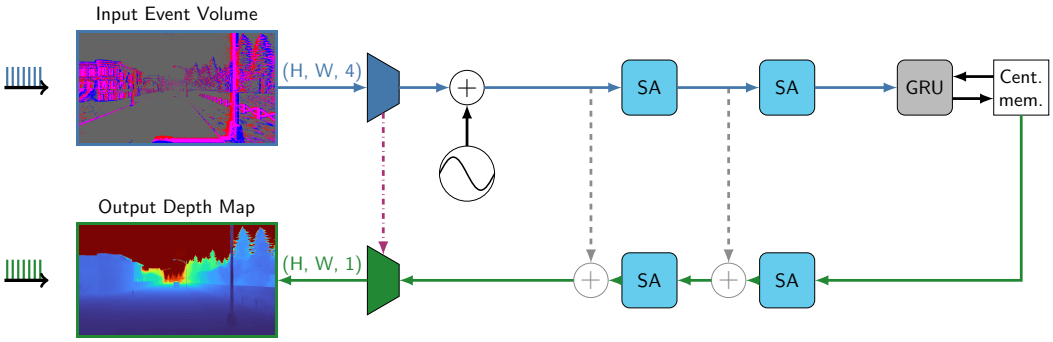

Figure 13. The alternative architecture without the LiDAR input, DELTA^{NL}.

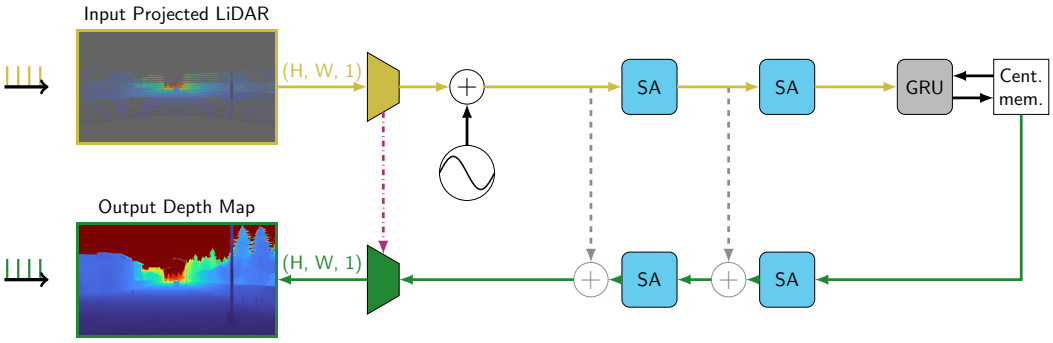

Figure 14. The alternative architecture without the event input, DELTA^{NE}.

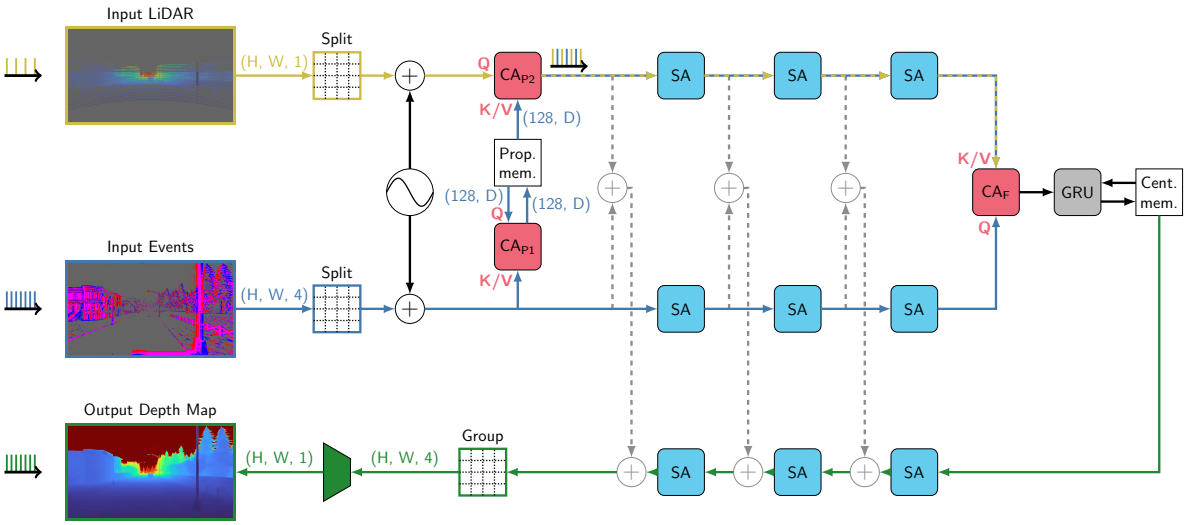

Figure 15. The alternative architecture where the convolutional encoding heads are replaced by a simple splitting into patches, DELTA$^{\text{NEH}}$.

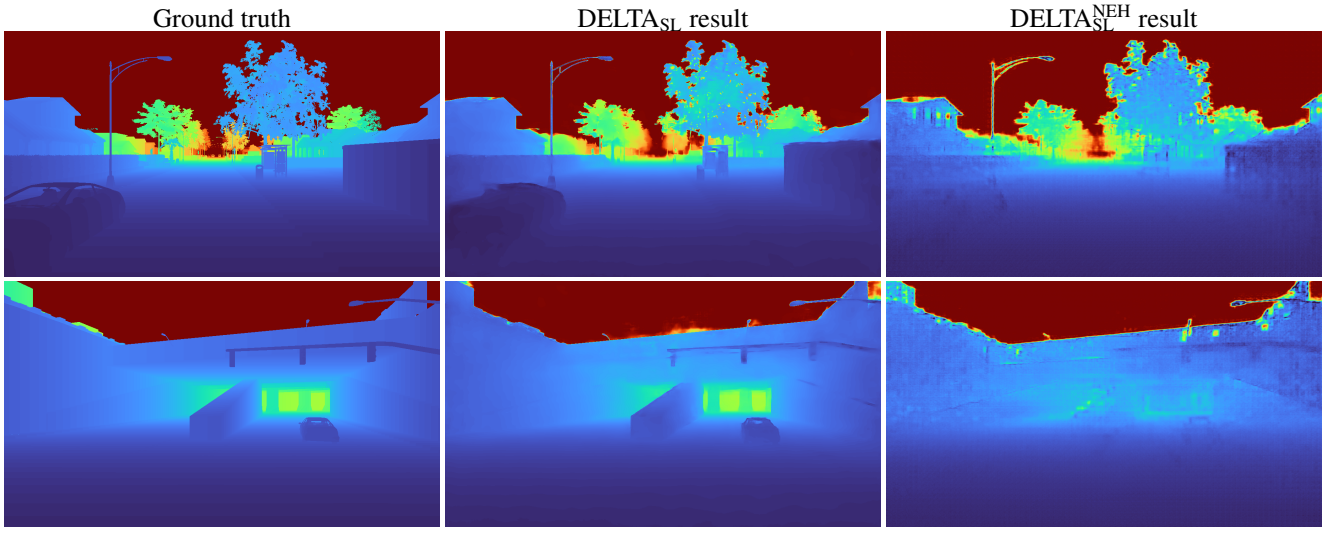

Figure 16. Results on the `Town01_08` (top) and `Town03_19` (bottom) sequences of SLED, for DELTA$_{SL}$ and DELTA$_{SL}^{NEH}$. Zoom on the numerical version may be required to better see the individual patches and artifacts for DELTA$_{SL}^{NEH}$.

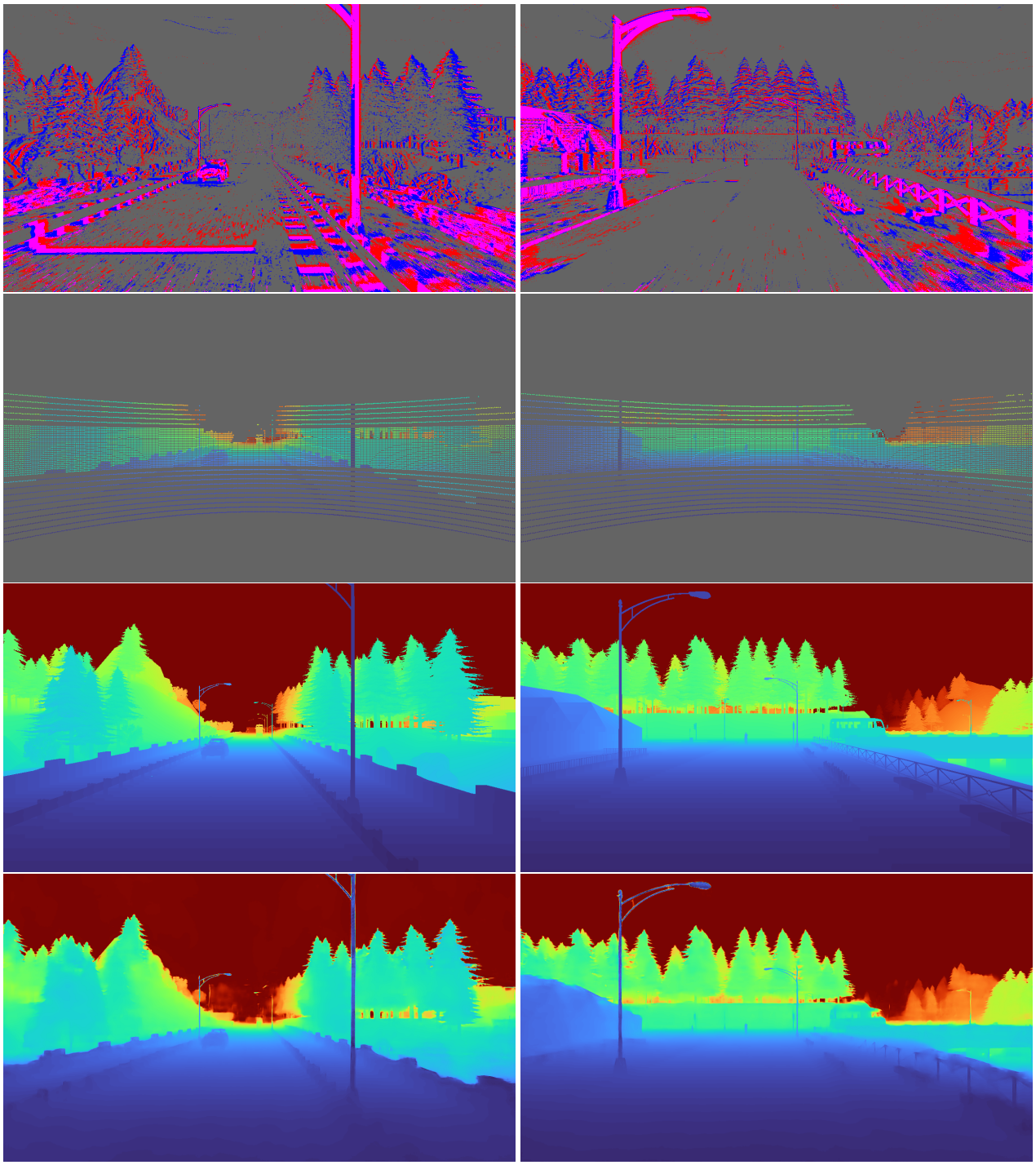

Figure 17. Additional results on the SLED dataset, on sequences `Town01_03` and `Town01_05`. From top to bottom: events, LiDAR projection, ground truth, our results.

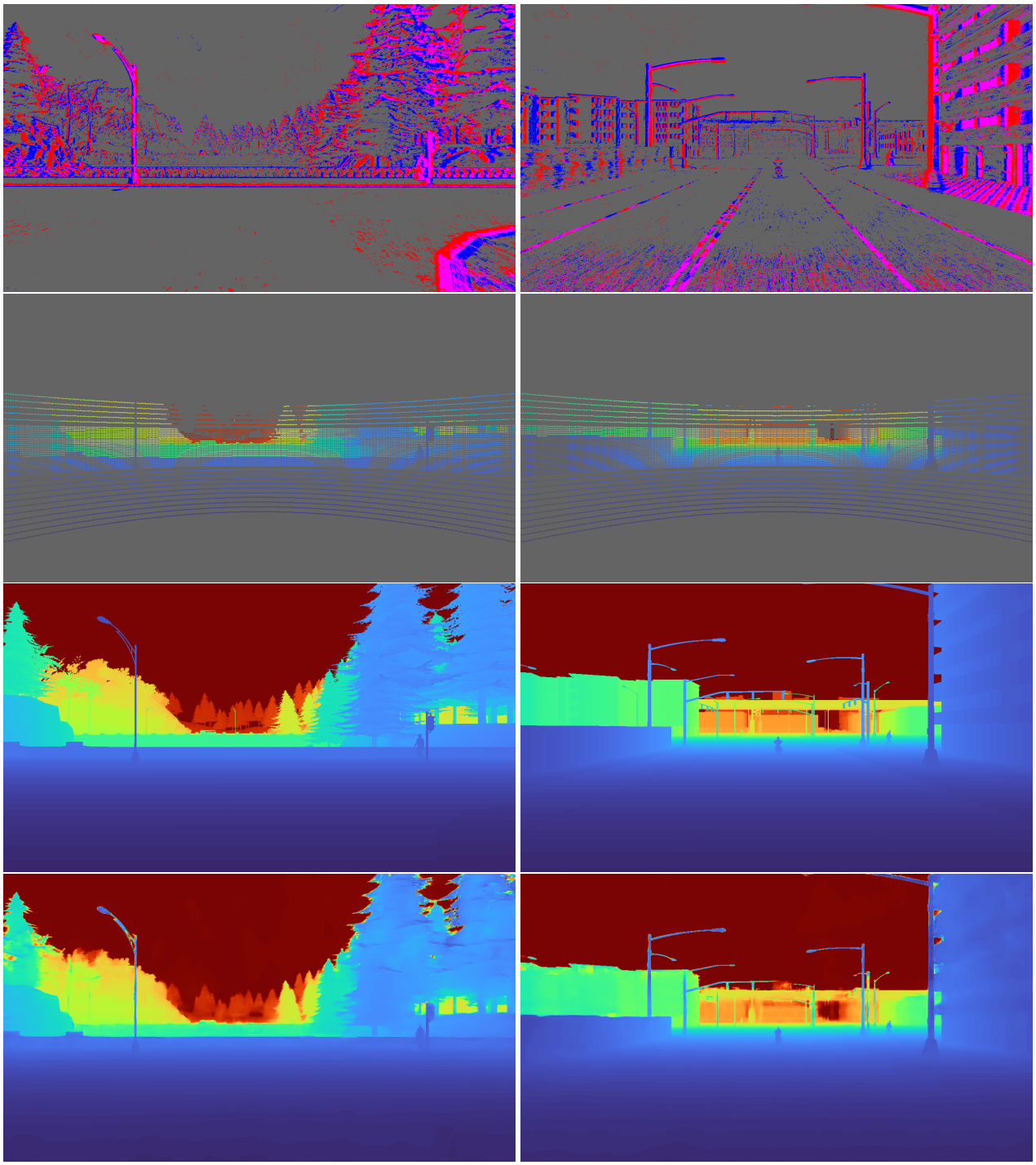

Figure 18. Additional results on the SLED dataset, on sequences `Town01_18` and `Town03_02`. From top to bottom: events, LiDAR projection, ground truth, our results.

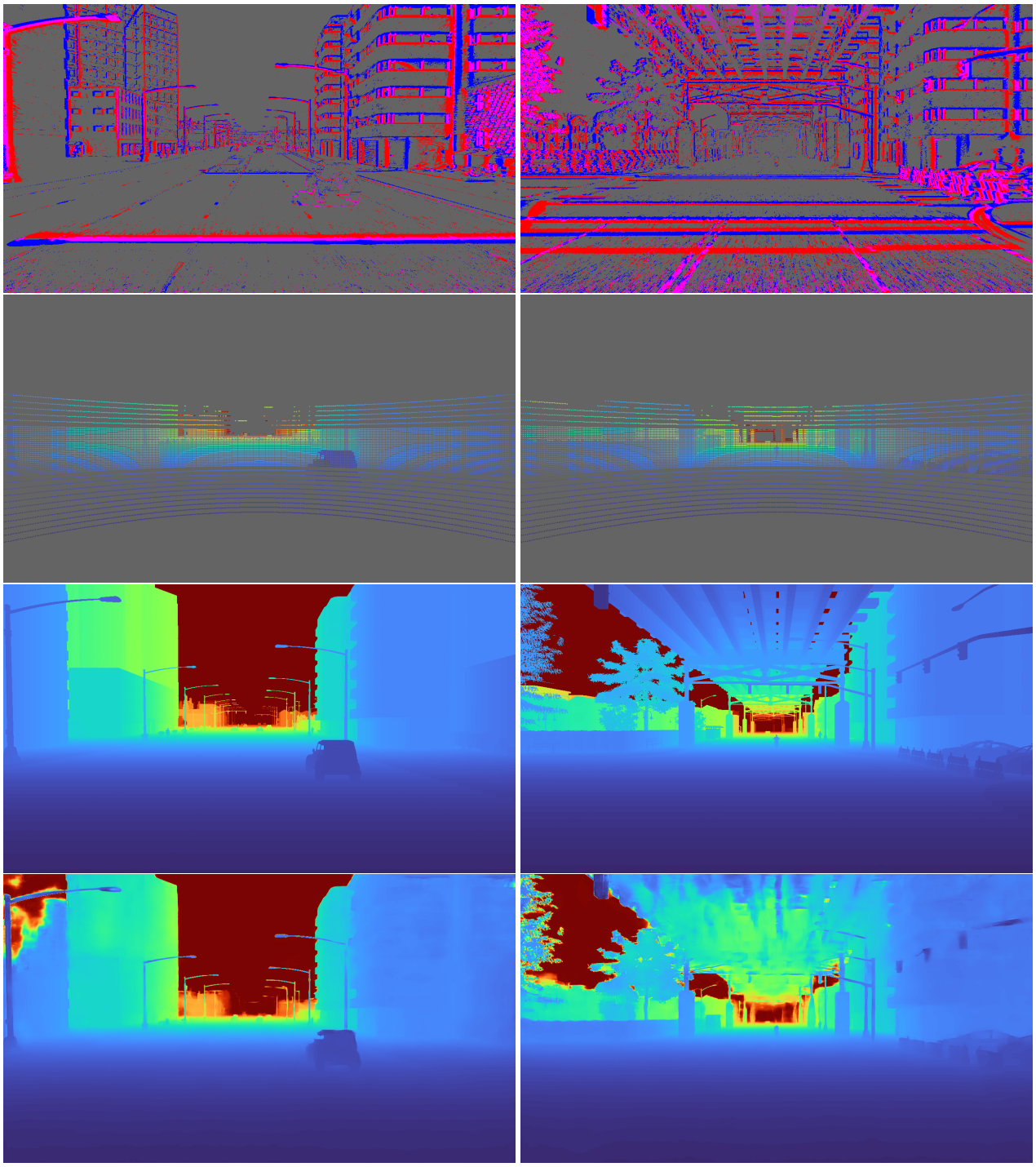

Figure 19. Additional results on the SLED dataset, on sequences `Town03_06` and `Town03_13`. From top to bottom: events, LiDAR projection, ground truth, our results. Shown here are two cases where DELTA_{SL} displays moderate to large errors for objects in the upper part of the depth maps (where no LiDAR data is available), like the building on the top left for the left column, and the suspended railway on the top for the right column.

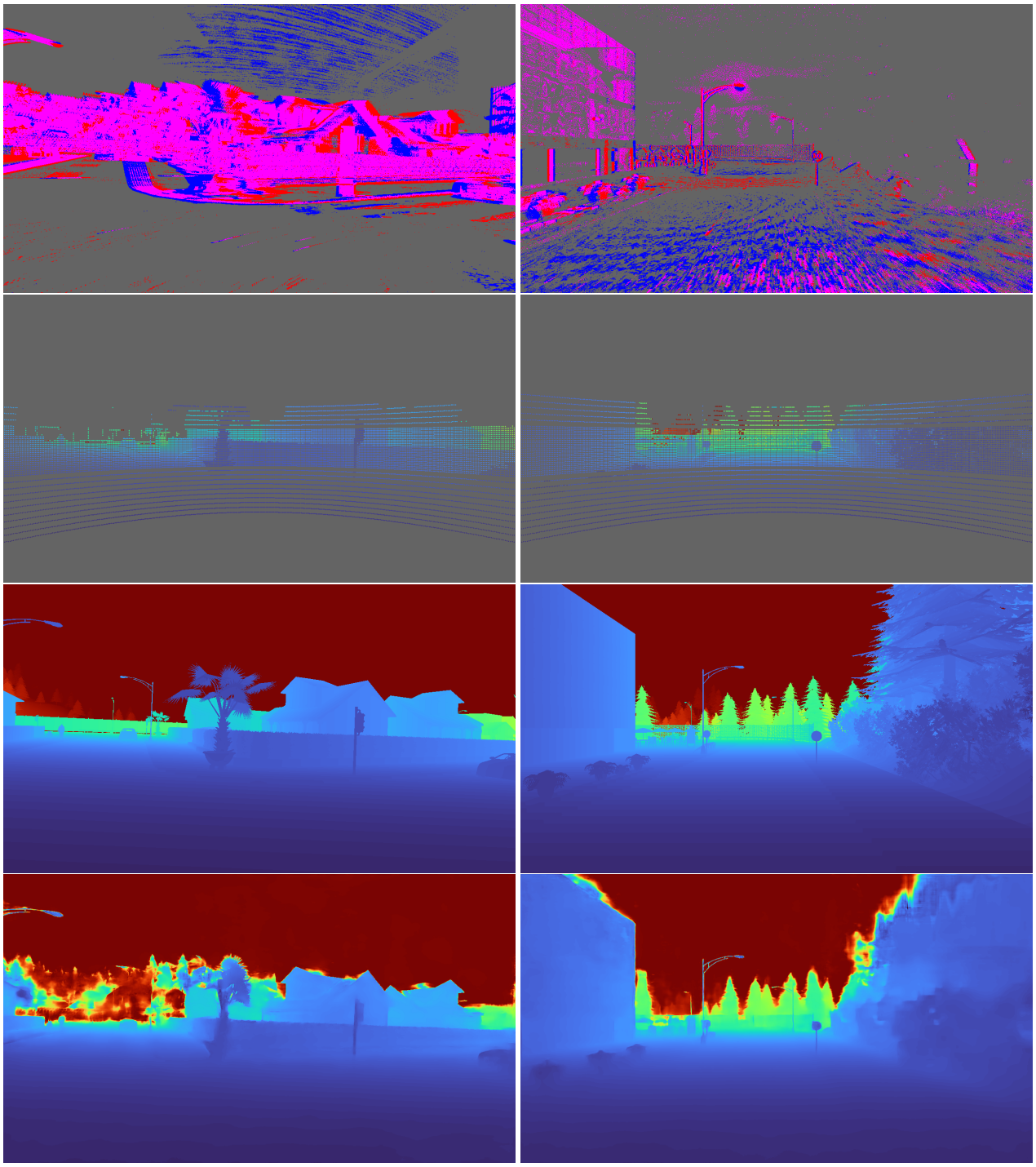

Figure 20. Additional results on the SLED dataset, on sequences `Town01_08` and `Town01_11`. From top to bottom: events, LiDAR projection, ground truth, our results. Shown here are two failure cases where DELTA$_{SL}$ displays large errors. Left: due to a high-speed sharp turn, a very high quantity of events is produced in the time window of accumulation, leading to information being lost in the event volume, and thus leading to an inaccurate depth estimation for background objects. Right: due to the limitations of the event camera in the CARLA simulator, dark objects in a night scene like the trees in the middle and on the right of the scene are not captured in the event stream of the SLED dataset, resulting in blurry depth estimations for these objects.

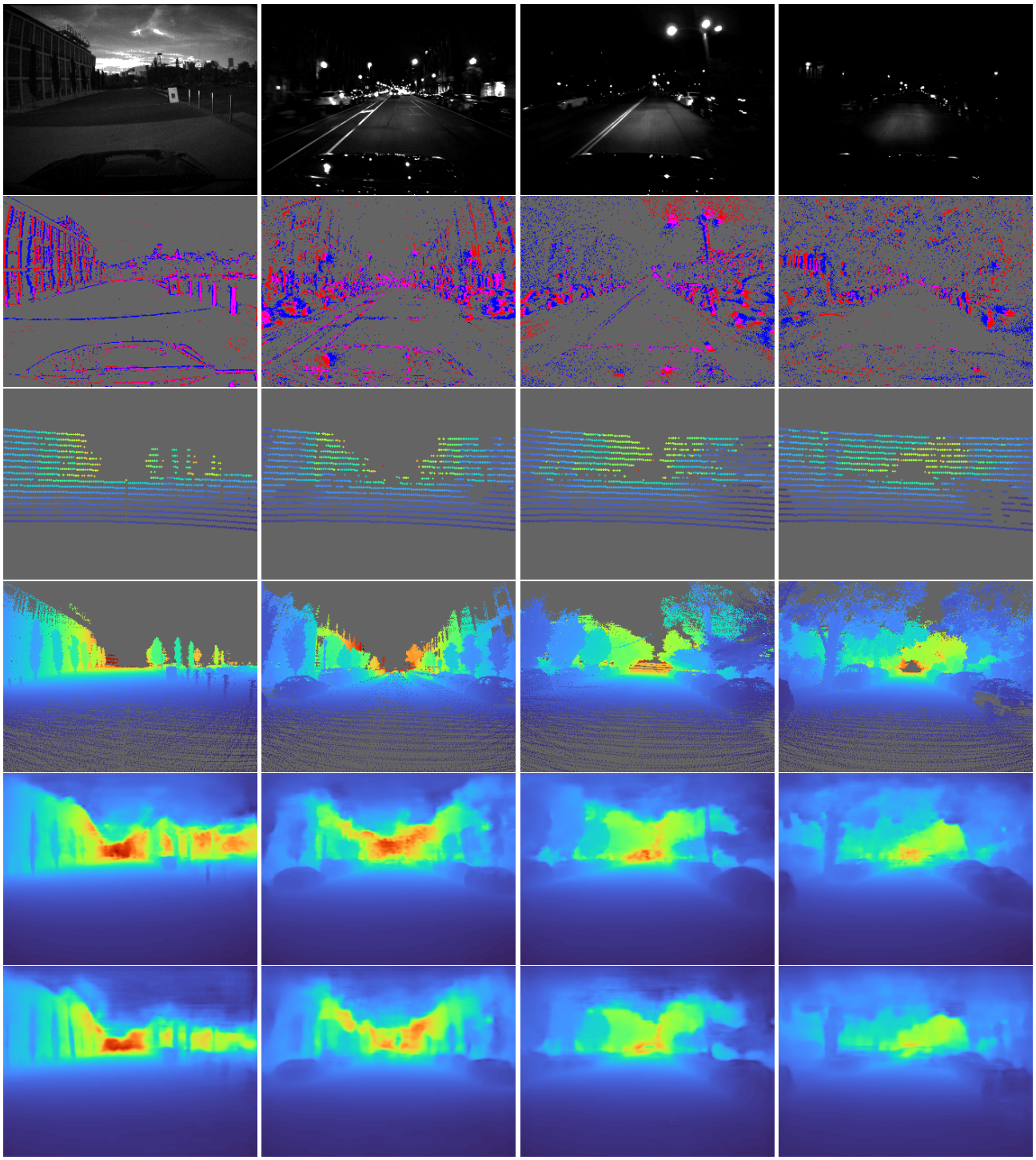

Figure 21. Additional results on the MVSEC dataset. Sequences shown, from left to right: `outdoor_day_1`; `outdoor_night_1`; `outdoor_night_2`; `outdoor_night_3`. From top to bottom: reference image of the scene; events; LiDAR projection (with size of points increased for a better visibility); ground truth; our results (DELTA_{MV}, $\text{DELTA}_{\text{SL}\rightarrow\text{MV}}$).

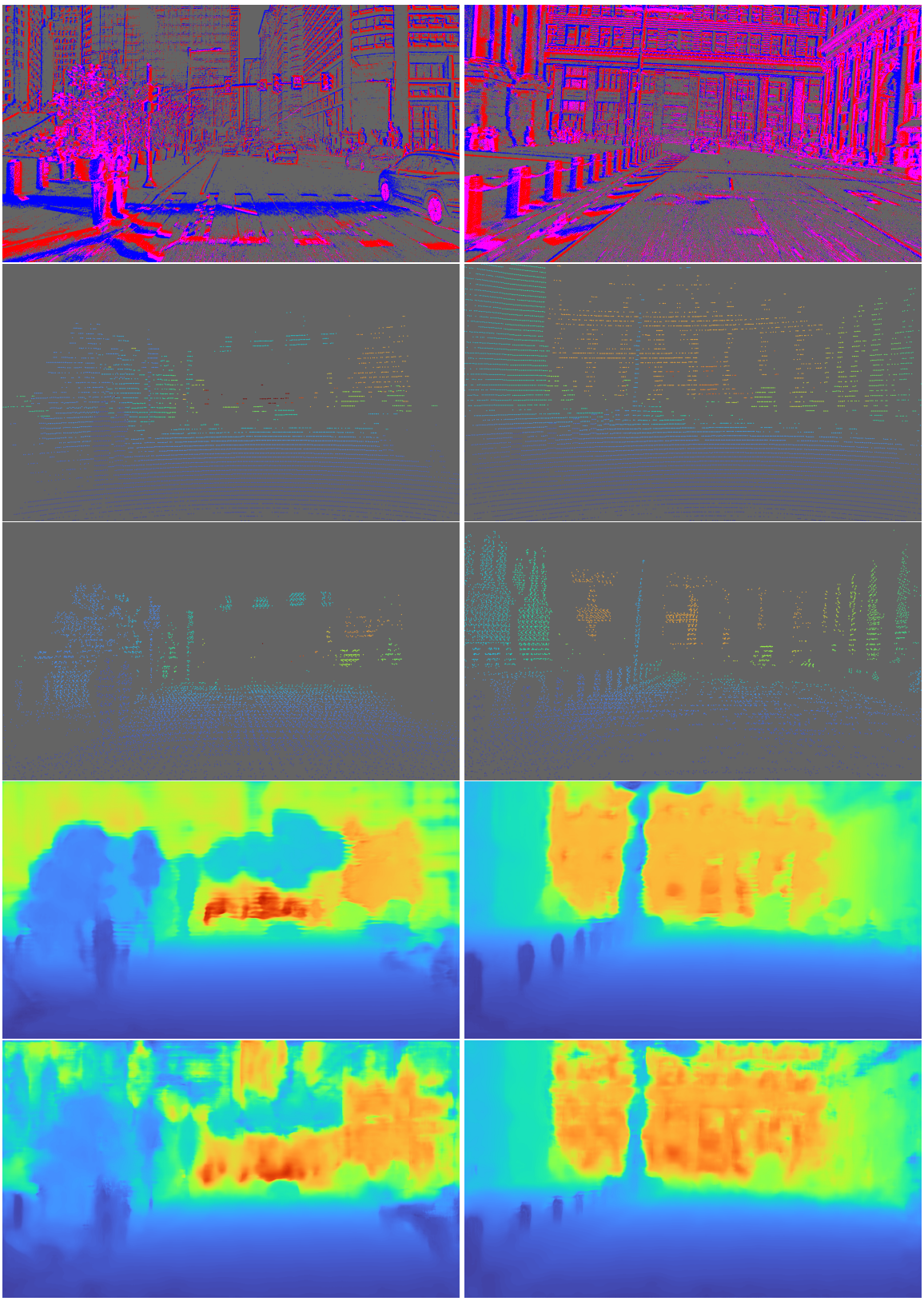

Figure 22. Additional results on the M3ED dataset, for the `city_hall_day` sequence. From top to bottom: events, LiDAR projection, ground truth, our results (DELTA_{M3}, $\text{DELTA}_{\text{SL}\rightarrow\text{M3}}$).

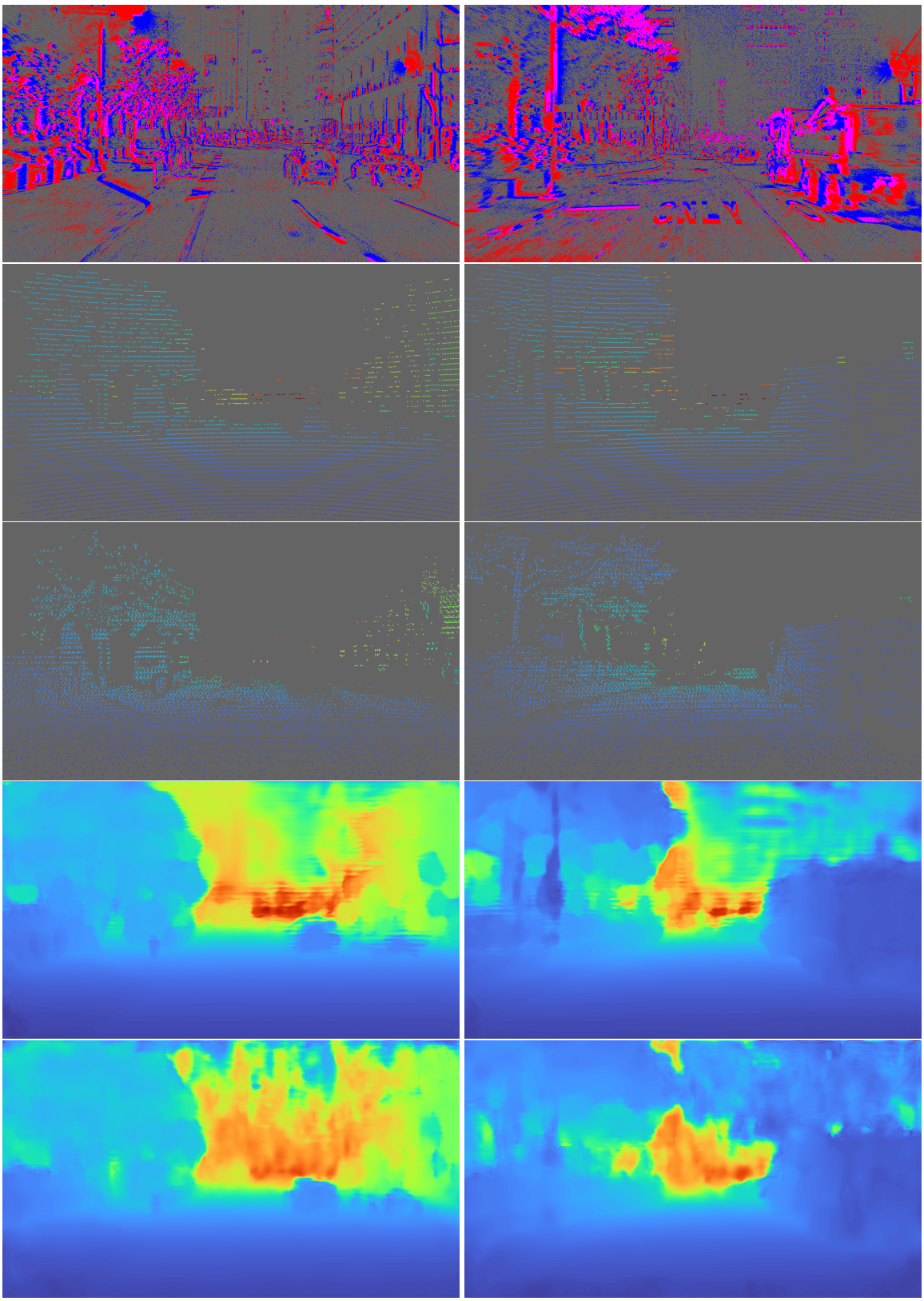

Figure 23. Additional results on the M3ED dataset, for the `city_hall_night` sequence. From top to bottom: events, LiDAR projection, ground truth, our results (DELTA_{M3}, $\text{DELTA}_{\text{SL}\rightarrow\text{M3}}$).