# Structure-Centric Robust Monocular Depth Estimation via Knowledge Distillation: Supplementary Material

In this supplementary material, we expand upon the main text by offering clarifications on its content, presenting a comprehensive set of experimental results, and including enhanced visual representations of feature extraction and depth estimation processes. This material serves to deepen the reader's understanding and provide a more thorough insight into the methodologies and outcomes discussed in the primary research.

## A Clarification on sub-problems

Our work addresses three sub-problems in monocular scene structure: *depth structure consistency*, *local texture disambiguation*, and *semantic and structural correlation*. Depth structure consistency is a fundamental problem that monocular depth estimation (MDE) needs to address. We clarify the issues of semantic and structure correlation that many previous works mentioned. Building on this foundation, we propose the depth disambiguation problem of local texture. Our main contributions are: observing the **influence of ambient lighting on scene structure** and using it to guide depth disambiguation of local texture; modeling the embedding **correlation between semantic features and scene structure features** through isomorphic graph knowledge distillation.

## B Trade-offs between clear and challenging performances

Further evaluation on the public datasets DENSE and DIODE confirms that our model outperforms baseline methods, which primarily address depth consistency, such as MonoDepth2 and MonoViT, in terms of accuracy on clear test sets. The comparative results are detailed in Tables 1 and 2.

## C Performance Metrics Detailed Explanation

In this section, we elaborate on the computation methodologies for all metrics utilized within our research, aiming to provide a comprehensive understanding of how performance is quantified. Our evaluation framework prioritizes lower values for Absolute Relative Difference (Abs Rel), Squared Relative Difference (Sq Rel), Root Mean Squared Error (RMSE), and Root Mean Squared Error in log scale (RMSE log), while higher values of $\delta_1$, $\delta_2$, and $\delta_3$ thresholds denote superior performance.

**Table 1: Generalization performance comparison of MDE on the DIODE benchmark.** To ensure fairness in comparisons and avoid introducing any corrupted samples, all the models mentioned above are trained on the KITTI Clear training dataset. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better. (SCDepthV3 employs an additional supervised depth estimation teacher model to provide labels.)

| Method | Knowledge | Abs Rel ↓ | Sq Rel ↓ | RMSE↓ | RMSE Log ↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.417 | 6.041 | 6.593 | 0.427 | 0.541 | 0.769 | 0.880 |
| SCDepthV1 | M | 0.416 | 5.807 | 6.196 | 0.430 | 0.518 | 0.765 | 0.886 |
| DynaDepth | M | 0.453 | 6.671 | 6.727 | 0.446 | 0.517 | 0.756 | 0.871 |
| HRDepth | M | 0.433 | 6.810 | 6.655 | 0.439 | 0.522 | 0.758 | 0.873 |
| SCDepthV3 | M+D | **0.365** | **4.370** | **5.526** | **0.386** | **0.576** | **0.808** | **0.907** |
| SGDepth | M+I | 0.459 | 7.409 | 7.160 | 0.459 | 0.503 | 0.741 | 0.863 |
| PackNet-SfM | M+S | 0.443 | 6.345 | 6.664 | 0.449 | 0.507 | 0.749 | 0.868 |
| **Ours$_{\mathrm{R}}$** | M+S | 0.414 | 6.011 | 6.381 | 0.426 | 0.542 | 0.771 | 0.882 |
| MonoViT | M | 0.348 | 3.750 | **5.006** | 0.367 | 0.599 | **0.825** | 0.911 |
| LiteMono | M | 0.454 | 6.598 | 6.724 | 0.465 | 0.486 | 0.734 | 0.855 |
| **Ours$_{\mathrm{T}}$** | M+S | **0.340** | **3.708** | 5.118 | **0.366** | 0.595 | **0.825** | **0.916** |

**Table 2:** Detailed performance comparison of monocular depth estimation on the DENSE Clear test set. To ensure fairness in comparisons and avoid introducing any corrupted samples, all the models mentioned above are trained on the KITTI Clear training dataset. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better.

| Method | Knowledge | Abs Rel↓ | Sq Rel↓ | RMSE↓ | RMSE Log↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.281 | 1.784 | 4.995 | 0.383 | 0.572 | 0.780 | 0.889 |
| SCDepthV1 | M | 0.297 | 1.822 | 5.147 | 0.417 | 0.507 | 0.724 | 0.869 |
| HRDepth | M | 0.272 | 1.501 | 4.681 | 0.366 | 0.532 | 0.785 | 0.923 |
| SCDepthV3 | M+D | 0.289 | 1.837 | 5.152 | 0.397 | 0.531 | 0.750 | 0.882 |
| DynaDepth | M+I | 0.273 | 1.506 | 4.912 | 0.386 | 0.493 | 0.774 | 0.900 |
| SGDepth | M+S | 0.261 | 1.459 | 4.856 | 0.368 | 0.547 | 0.794 | 0.911 |
| PackNet-SfM | M+S | 0.258 | 1.369 | 4.585 | 0.357 | 0.550 | 0.800 | 0.917 |
| **Ours$_{\mathbf{R}}$** | M+S | 0.256 | 1.445 | 4.645 | 0.354 | 0.575 | 0.815 | 0.911 |
| MonoViT | M | 0.242 | 1.152 | 4.320 | 0.330 | 0.555 | 0.814 | 0.944 |
| LiteMono | M | 0.310 | 2.096 | 5.095 | 0.367 | 0.481 | 0.802 | 0.921 |
| **Ours$_{\mathbf{T}}$** | M+S | 0.218 | 0.915 | 3.725 | 0.298 | 0.606 | 0.849 | 0.961 |

**Absolute Relative Difference.** The Absolute Relative Difference quantifies the average relative difference between the estimated depth values and the ground truth, defined as

$$\mathcal{E}_{\text{AbsRel}} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{D}_t^i - D_t^i|}{D_t^i}, \tag{1}$$

where $D_t^i$ represents the ground truth depth for the $i^{th}$ pixel, $\hat{D}_t^i$ is the estimated depth, and $N$ denotes the total number of instances in the dataset.

**Squared Relative Difference.** The Squared Relative Difference emphasizes larger errors more significantly by computing the average of the squared relative differences between the predicted and actual depth values as

$$\mathcal{E}_{\text{SqRel}} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\hat{D}_t^i - D_t^i)^2}{D_t^i}. \tag{2}$$

**Root Mean Squared Error.** The RMSE is a standard metric for evaluating the accuracy of predictions, calculating the square root of the average of the squared differences between estimated and true values:

$$\mathcal{E}_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{D}_t^i - D_t^i)^2}. \tag{3}$$

**Root Mean Squared Error Log.** The RMSE log applies a logarithmic transformation prior to computing the squared differences, calculated as

$$\mathcal{E}_{\text{RMSELog}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(\hat{D}_t^i + 1) - \log(D_t^i + 1))^2}. \tag{4}$$

**Accuracy under Thresholds.** These thresholds measure the percentage of predictions that fall within specified factor thresholds of the actual depth values, aiming to capture the accuracy of the depth estimation, defined as:

$$\delta_r = \frac{1}{N}\sum_{i=1}^{N}\left(\max\left(\frac{D_t^i}{\hat{D}_t^i}, \frac{\hat{D}_t^i}{D_t^i}\right) < 1.25^r\right), \tag{5}$$

where $r \in \{1, 2, 3\}$, corresponding to the thresholds $1.25, 1.25^2$, and $1.25^3$ respectively. A prediction is considered accurate if its ratio to the ground truth is less than the threshold value.

By leveraging these metrics, we ensure a holistic evaluation of our model's performance, capturing both the magnitude and distribution of the errors, as well as the precision of the depth estimations.

**Table 3:** Knowledge Form Abbreviations

| Abbr. | Knowledge Form |
|---|---|
| M | Monocular Camera |
| D | Pseudo Depth Map |
| I | Inertial Measurement Unit |
| S | Implicit & Explicit Semantics |

**Table 4:** The weights for balancing the losses, including $\lambda_p$, $\lambda_v$, $\lambda_e$, $\lambda_r$, $\lambda_o$, and $\lambda_d$.

| $\lambda$ | Value |
|---|---|
| $\lambda_p$, $\lambda_e$ | 1.0 |
| $\lambda_v$, $\lambda_r$ | 0.1 |
| $\lambda_o$, $\lambda_d$ | 0.001 |

## D Overview of Additional Knowledge Forms

In our experimental comparison tables, we annotate the types of knowledge forms utilized by the methods, with the detailed descriptions of these knowledge forms presented in Table 3. Our method uniquely uses implicit semantic information, unlike other related works that depend on explicitly annotated semantic labels for training constraints.

## E Weights of Losses

The table in 4 shows the specific values for $\lambda_p$, $\lambda_v$, $\lambda_e$, $\lambda_r$, $\lambda_o$, and $\lambda_d$ that we employ. Our primary objective remains to guide the model in achieving depth estimation, hence we prioritize the weights for depth structure consistency. To further guide the estimation of illumination, we set these weights higher than those for feature decoupling or distillation loss. Setting overly large weights on losses that affect features can disrupt the effective training of depth estimation.

## F Detailed Performace Comparion

In the main text, due to space constraints, we have selected only a subset of the most representative metrics. For the sake of completeness, we provide a detailed performance comparison of all metrics in the supplementary material.

## G Encoder Feature Visualization Techniques

As displayed in Algorithm 1, the visualization $V_t$ emerges as a powerful tool to demystify the complex feature representations generated by the texture encoder and structure encoder in . By leveraging Principal Component Analysis (PCA) to distill the multidimensional feature space into a singular, interpretable dimension. This method effectively highlights the spatial distribution and intensity variations within the encoded features, offering insights into the encoder's focus and its differentiation of input data aspects. The transition from high-dimensional representations to a comprehensible visual format not only aids in

**Table 5:** Detailed performance comparison of monocular depth estimation on the KITTI Snow test set. To ensure fairness in comparisons and avoid introducing any corrupted samples, all the models mentioned above are trained on the KITTI Clear training dataset. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better.

| Method | Knowledge | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE Log ↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.462 | 6.309 | 11.397 | 0.537 | 0.330 | 0.608 | 0.796 |
| SCDepthV1 | M | 0.435 | 4.962 | 10.871 | 0.528 | 0.335 | 0.619 | 0.804 |
| HRDepth | M | 0.460 | 5.826 | 10.749 | 0.538 | 0.331 | 0.605 | 0.790 |
| SCDepthV3 | M+D | 0.410 | 4.159 | 10.721 | 0.523 | 0.330 | 0.619 | 0.816 |
| DynaDepth | M+I | 0.484 | 6.179 | 11.449 | 0.571 | 0.301 | 0.571 | 0.768 |
| SGDepth | M+S | 0.473 | 6.487 | 11.451 | 0.541 | 0.322 | 0.606 | 0.790 |
| PackNet-SfM | M+S | 0.426 | 5.372 | 10.879 | 0.509 | 0.349 | 0.638 | 0.820 |
| **Ours$_{\mathbf{R}}$** | M+S | 0.404 | 4.454 | 10.093 | 0.490 | 0.358 | 0.660 | 0.841 |
| LiteMono | M | 0.381 | 3.858 | 9.794 | 0.476 | 0.355 | 0.675 | 0.859 |
| MonoViT | M | 0.230 | 1.893 | 6.951 | 0.316 | 0.621 | 0.858 | 0.942 |
| **Ours$_{\mathbf{T}}$** | M+S | 0.224 | 1.790 | 6.899 | 0.314 | 0.627 | 0.859 | 0.942 |

**Table 6:** Detailed performance comparison of monocular depth estimation on the KITTI Frost test set. To ensure fairness in comparisons and avoid introducing any corrupted samples, all the models mentioned above are trained on the KITTI Clear training dataset. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better.

| Method | Knowledge | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE Log ↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.289 | 2.926 | 8.181 | 0.376 | 0.534 | 0.795 | 0.908 |
| SCDepthV1 | M | 0.259 | 2.284 | 7.803 | 0.369 | 0.572 | 0.809 | 0.911 |
| HRDepth | M | 0.353 | 3.719 | 9.007 | 0.455 | 0.452 | 0.703 | 0.853 |
| SCDepthV3 | M+D | 0.321 | 2.897 | 9.223 | 0.432 | 0.436 | 0.741 | 0.887 |
| DynaDepth | M+I | 0.320 | 3.307 | 8.600 | 0.424 | 0.493 | 0.747 | 0.878 |
| SGDepth | M+S | 0.265 | 2.516 | 7.573 | 0.349 | 0.575 | 0.821 | 0.923 |
| PackNet-SfM | M+S | 0.315 | 3.357 | 8.704 | 0.398 | 0.491 | 0.767 | 0.894 |
| **Ours$_{\mathbf{R}}$** | M+S | 0.236 | 1.994 | 7.266 | 0.326 | 0.608 | 0.855 | 0.939 |
| LiteMono | M | 0.258 | 2.152 | 7.336 | 0.331 | 0.567 | 0.835 | 0.940 |
| MonoViT | M | 0.212 | 1.759 | 6.717 | 0.289 | 0.663 | 0.881 | 0.954 |
| **Ours$_{\mathbf{T}}$** | M+S | 0.202 | 1.641 | 6.614 | 0.285 | 0.678 | 0.887 | 0.954 |

**Table 7:** Detailed performance comparison of monocular depth estimation on the KITTI Motion Blur test set. To ensure fairness in comparisons and avoid introducing any corrupted samples, all the models mentioned above are trained on the KITTI Clear training dataset. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better.

| Method | Knowledge | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE Log ↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.242 | 2.209 | 7.613 | 0.343 | 0.613 | 0.833 | 0.926 |
| SCDepthV1 | M | 0.334 | 3.530 | 8.972 | 0.462 | 0.492 | 0.728 | 0.848 |
| HRDepth | M | 0.345 | 3.599 | 9.314 | 0.482 | 0.476 | 0.708 | 0.833 |
| SCDepthV3 | M+D | 0.282 | 2.630 | 8.834 | 0.396 | 0.537 | 0.777 | 0.892 |
| DynaDepth | M+I | 0.345 | 3.654 | 9.167 | 0.455 | 0.487 | 0.715 | 0.848 |
| SGDepth | M+S | 0.259 | 2.305 | 7.886 | 0.371 | 0.578 | 0.811 | 0.909 |
| PackNet-SfM | M+S | 0.291 | 2.725 | 8.649 | 0.402 | 0.519 | 0.770 | 0.894 |
| **Ours$_{\mathbf{R}}$** | M+S | 0.238 | 2.087 | 7.447 | 0.338 | 0.617 | 0.840 | 0.931 |
| LiteMono | M | 0.252 | 2.094 | 7.555 | 0.345 | 0.582 | 0.825 | 0.926 |
| MonoViT | M | 0.189 | 1.442 | 6.651 | 0.275 | 0.699 | 0.896 | 0.961 |
| **Ours$_{\mathbf{T}}$** | M+S | 0.183 | 1.416 | 6.645 | 0.272 | 0.710 | 0.899 | 0.961 |

**Table 8:** Detailed performance comparison of monocular depth estimation on the KITTI Clear test set. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better.

| Method | Knowledge | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE Log ↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.117 | 0.957 | 4.953 | 0.195 | 0.875 | 0.958 | 0.980 |
| SCDepthV1 | M | 0.118 | 0.870 | 4.964 | 0.195 | 0.860 | 0.957 | 0.982 |
| HRDepth | M | 0.106 | 0.815 | 4.598 | 0.184 | 0.891 | 0.963 | 0.982 |
| SCDepthV3 | M+D | 0.117 | 0.735 | 4.682 | 0.186 | 0.866 | 0.961 | 0.984 |
| DynaDepth | M+I | 0.114 | 0.890 | 4.914 | 0.194 | 0.876 | 0.959 | 0.981 |
| SGDepth | M+S | 0.117 | 0.930 | 4.897 | 0.195 | 0.874 | 0.958 | 0.980 |
| PackNet-SfM | M+S | 0.116 | 0.872 | 4.970 | 0.198 | 0.868 | 0.956 | 0.980 |
| **Ours$_{\mathbf{R}}$** | M+S | 0.118 | 0.872 | 4.869 | 0.196 | 0.869 | 0.958 | 0.981 |
| LiteMono | M | 0.113 | 0.910 | 4.874 | 0.193 | 0.882 | 0.960 | 0.980 |
| MonoViT | M | 0.100 | 0.729 | 4.410 | 0.176 | 0.898 | 0.967 | 0.984 |
| **Ours$_{\mathbf{T}}$** | M+S | 0.105 | 0.796 | 4.515 | 0.181 | 0.894 | 0.964 | 0.982 |

**Table 9:** Detailed performance comparison of monocular depth estimation on the DENSE Fog test set. To ensure fairness in comparisons and avoid introducing any corrupted samples, all the models mentioned above are trained on the KITTI Clear training dataset. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better.

| Method | Knowledge | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE Log ↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.266 | 1.484 | 4.865 | 0.374 | 0.526 | 0.804 | 0.906 |
| SCDepthV1 | M | 0.269 | 1.600 | 5.125 | 0.410 | 0.539 | 0.751 | 0.866 |
| HRDepth | M | 0.262 | 1.402 | 4.707 | 0.366 | 0.533 | 0.797 | 0.920 |
| SCDepthV3 | M+D | 0.257 | 1.568 | 5.087 | 0.380 | 0.578 | 0.788 | 0.892 |
| DynaDepth | M+I | 0.271 | 1.589 | 5.191 | 0.408 | 0.519 | 0.776 | 0.891 |
| SGDepth | M+S | 0.260 | 1.478 | 4.967 | 0.379 | 0.548 | 0.801 | 0.909 |
| PackNet-SfM | M+S | 0.254 | 1.346 | 4.697 | 0.360 | 0.532 | 0.812 | 0.918 |
| **Ours$_{\mathbf{R}}$** | M+S | 0.240 | 1.262 | 4.567 | 0.342 | 0.580 | 0.834 | 0.924 |
| MonoViT | M | 0.239 | 1.223 | 4.600 | 0.345 | 0.556 | 0.824 | 0.926 |
| LiteMono | M | 0.325 | 1.922 | 4.863 | 0.383 | 0.422 | 0.727 | 0.926 |
| **Ours$_{\mathbf{T}}$** | M+S | 0.224 | 1.032 | 3.907 | 0.312 | 0.583 | 0.848 | 0.948 |

**Table 10:** Detailed performance comparison of monocular depth estimation on the DENSE Rain test set. To ensure fairness in comparisons and avoid introducing any corrupted samples, all the models mentioned above are trained on the KITTI Clear training dataset. Upper Section: ResNet-based Architectures. Lower Section: Hybrid Architectures. ↓: Lower is better. ↑: Higher is better.

| Method | Knowledge | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE Log ↓ | $\delta_1$ ↑ | $\delta_2$ ↑ | $\delta_3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| MonoDepth2 | M | 0.274 | 1.554 | 4.962 | 0.381 | 0.517 | 0.792 | 0.901 |
| SCDepthV1 | M | 0.266 | 1.570 | 5.118 | 0.396 | 0.537 | 0.765 | 0.882 |
| HRDepth | M | 0.262 | 1.394 | 4.730 | 0.366 | 0.528 | 0.804 | 0.918 |
| SCDepthV3 | M+D | 0.253 | 1.513 | 5.087 | 0.377 | 0.581 | 0.797 | 0.894 |
| DynaDepth | M+I | 0.281 | 1.657 | 5.311 | 0.415 | 0.504 | 0.771 | 0.886 |
| SGDepth | M+S | 0.252 | 1.413 | 4.912 | 0.372 | 0.563 | 0.806 | 0.911 |
| PackNet-SfM | M+S | 0.256 | 1.370 | 4.765 | 0.365 | 0.533 | 0.811 | 0.913 |
| **Ours$_{\mathbf{R}}$** | M+S | 0.247 | 1.355 | 4.642 | 0.348 | 0.587 | 0.821 | 0.918 |
| MonoViT | M | 0.237 | 1.280 | 4.767 | 0.354 | 0.564 | 0.822 | 0.914 |
| LiteMono | M+S | 0.322 | 1.877 | 4.787 | 0.378 | 0.414 | 0.757 | 0.928 |
| **Ours$_{\mathbf{T}}$** | M+S | 0.224 | 1.083 | 4.046 | 0.321 | 0.571 | 0.841 | 0.941 |

qualitative analysis but also enhances our understanding of the encoder's operational nuances. Thus, $V_t$ serves as a bridge, translating abstract numerical data into a visual narrative that succinctly conveys the encoder's learning dynamics and its interpretative power over the input data.

**Algorithm 1** Visualization of Encoder-Generated Features

**Require:** $F_t$: The set of features generated from image $I_t$ by an encoder, where $F_t[c]$ represents the feature of channel $c$.
**Ensure:** $V_t$: The visualization of the encoder's feature representations.
Initialize an empty list $L_{\text{interp}}$ to hold interpolated features for visualization.
**for** each channel $c$ in $F_t$ **do**
Resize $F_t[c]$ to a predefined shape using bilinear interpolation and add to $L_{\text{interp}}$.
**end for**
$U_t \leftarrow$ Concatenate all features in $L_{\text{interp}}$ along the feature dimension.
Flatten $U_t$ spatially into a 2D array and transpose it, making features as rows.
Apply PCA to reduce the dimensionality of the transposed array to one dimension.
Compute the lower and upper bounds as the 5th and 95th percentiles of the PCA-reduced array, respectively.
Clip the PCA-reduced array within these bounds.
Normalize the normalized array to the range [0, 255] and convert it to an unsigned 8-bit gray-scale image.
Reshape the scaled array back to the original shape of $I_t$ to obtain the visualization $V_t$.

## H Detailed Feature Visualization Results

We provide representative feature visualization results in the main text. In the supplementary material, we present feature visualizations in a broader range of scenarios to demonstrate the generalizability of our feature interpretability in Figure 1.

## I Visualization of Depth

We visualize the estimated depth results on an image with a complex scene from the KITTI test sequence as displayed in Figure 2, using different models including MonoViT, LiteMono, HRDepth, and SGDepth. The visualizations show $\text{Ours}_{\text{T}}$ can identify the overhead bridge in dark areas for clear scenes, while experiencing less scene interference compared to other methods.

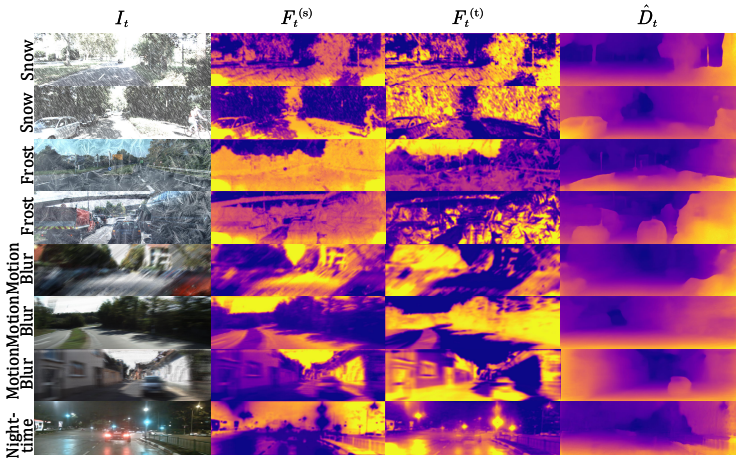


**Fig. 1:** Detailed visualization of intermediate feature maps. For features $F_t^{(s)}$ and $F_t^{(t)}$, yellow indicates high values. For predicted depth $\hat{D}_t$, purple denotes high values. We provide a detailed description of the computation method for visual feature maps in the supplementary materials.

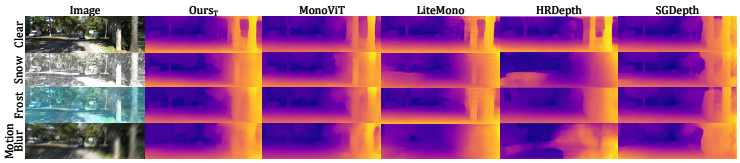


**Fig. 2:** Visualized depth comparison of Ours$_{\mathrm{T}}$, MonoViT, LiteMono, HRDepth, and SGDepth.