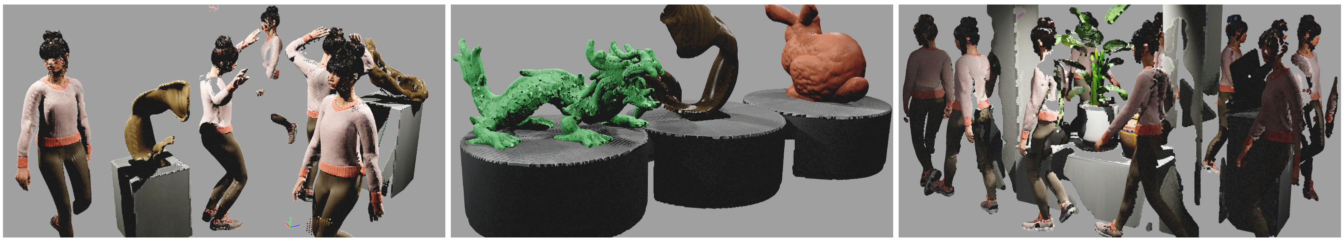
TemPCC: Supplementary Material

(a) Training Scene (Unreal Engine 5) (b) Validation Scene A (Unreal Engine 5) (c) Validation Scene B (Unreal Engine 5)



(d) PC captured in Training Scene (e) PC captured in Validation Scene A (f) PC captured in Validation Scene B

Figure 1: Our dataset comprises the following three scenes: (1) a training scene featuring people walking around and objects rotating at various speeds, illustrated in parts (a) and (d); (2) Validation Scene A with slowly rotating objects, illustrated in parts (b) and (e); and (3) a validation scene with many people walking in circles, illustrated in parts (c) and (f). Virtual RGB-D cameras are visualized in blue in (a-c).

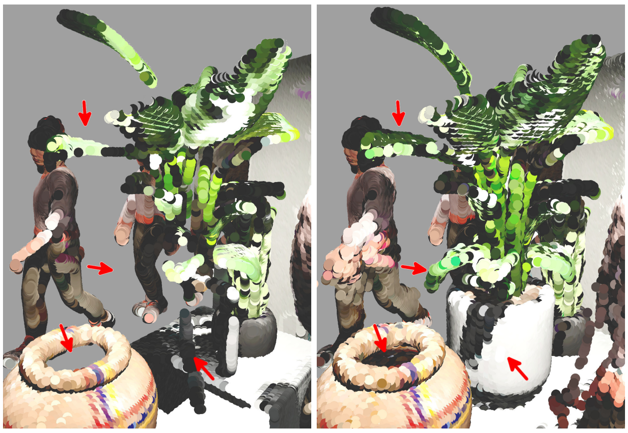


Figure 2: Plant from Validation Scene B captured by three cameras, shown partly occluded on the left and temporally completed on the right. Static objects are well completed during occlusion.

Figure 3: Validation Scene B, shown partly occluded on the left and temporally completed on the right, captured by three cameras. It is evident that especially smaller occlusions are effectively completed (e.g. the plant, the walls and some self occlusions by the animated humans). However, minor artifacts may occur on the limbs and bodies of the animated characters, resulting from their occasionally very rapid movements.

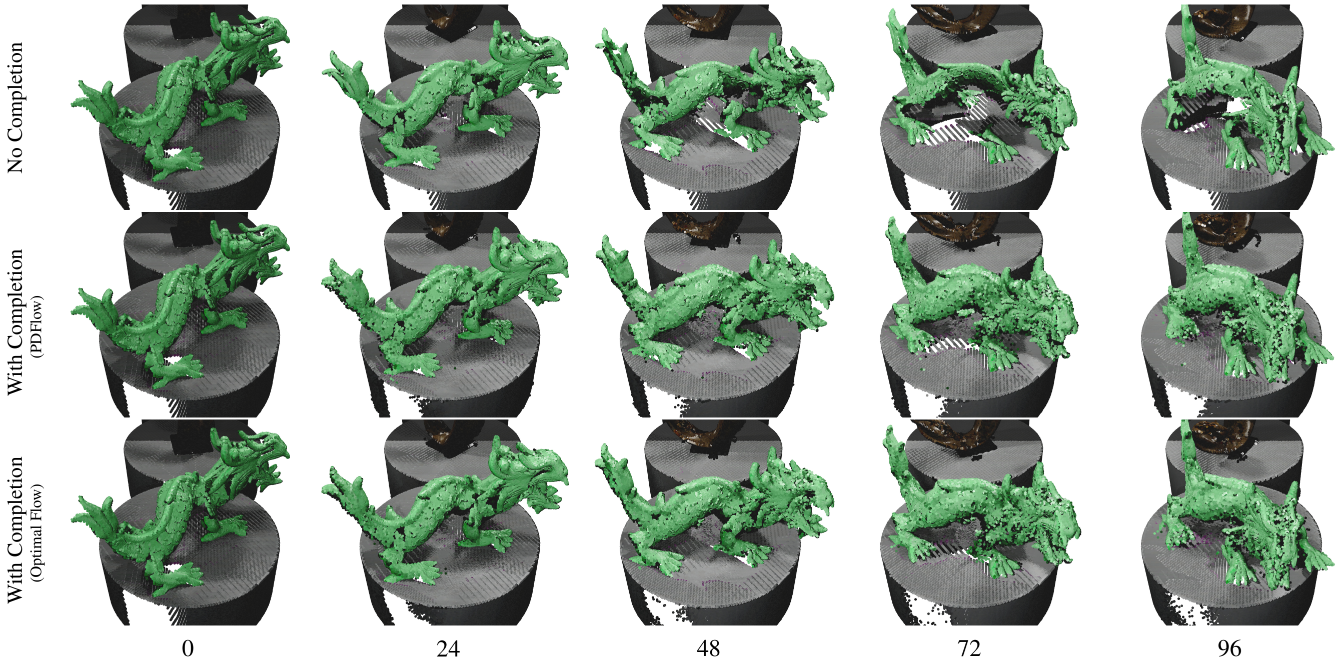


Figure 4: A rigid dragon from Validation Scene A, slowly rotating and captured by three cameras. This is a temporal sequence from frames 0, 24, 48, 72, and 96; the top shows the incomplete point cloud, and the bottom ones show the completed point clouds using different image flow techniques. Notably, the body of the dragon is well captured and reconstructed during the temporal completion. *For visualization purposes, the scene has been cropped and rendered on a white background.*

Table 1: Divergence of occluded points in comparison to the ground truth path length, exactly 30 frames after occlusion started, using a network trained with Optimal Flow (Opt) or PDFlow (PD) as input, while using Optimal Flow (Opt) or PDFlow (PD) as image flow technique. The term D_{GTPath} denotes the ground truth distance that an occluded point has traveled after 30 frames, measured in meters. The term D_{CoAbs} refers to the error between the completed point and the corresponding ground truth point after the same duration, also measured in meters. The ratio of these values, denoted as 'Rel.', effectively quantifies the relative error. $|P_{\text{Occ}}|$ denotes the number of occluded points considered. It is evident that a network trained with optimal flow achieves good results when using optimal flow (see Opt+Opt). However, when the network trained on optimal flow is directly applied using PDFlow, the error significantly increases in all scenarios (see Opt+PD). If TinyFlowNet is instead trained directly on the flow calculated by PDFlow for the visible points (see PD+PD), the error is only marginally higher on average compared to when using optimal flow exclusively.

	Training Scene				**Validation Scene A**				**Validation Scene B**			
Net + Flow / Fr.	D_{CoAbs}	D_{GTPath}	Rel.	$\lvert P_{\text{Occ}}\rvert$	D_{CoAbs}	D_{GTPath}	Rel.	$\lvert P_{\text{Occ}}\rvert$	D_{CoAbs}	D_{GTPath}	Rel.	$\lvert P_{\text{Occ}}\rvert$
Opt + Opt / 30	0.039	0.301	12.94 %	0.345 M	0.023	0.148	15.55 %	0.073 M	0.075	0.45	16.74 %	0.353 M
Opt + PD / 30	0.082	0.273	29.92 %	0.289 M	0.05	0.139	35.7 %	0.076 M	0.083	0.132	62.95 %	0.82 M
PD + PD / 30	0.045	0.306	14.8 %	0.338 M	0.029	0.15	19.15 %	0.075 M	0.077	0.346	22.33 %	0.611 M

Table 2: Impact of the number of cameras on completion error. It is evident that using multiple cameras results in lower error rates.

	Validation Scene A			
	D_{CoAbs}	D_{GTPath}	Rel.	$\lvert P_{\text{Occ}}\rvert$
1 Cameras	0.033	0.155	21.43 %	0.053 M
2 Cameras	0.03	0.161	18.43 %	0.063 M
3 Cameras	0.023	0.148	15.55 %	0.073 M

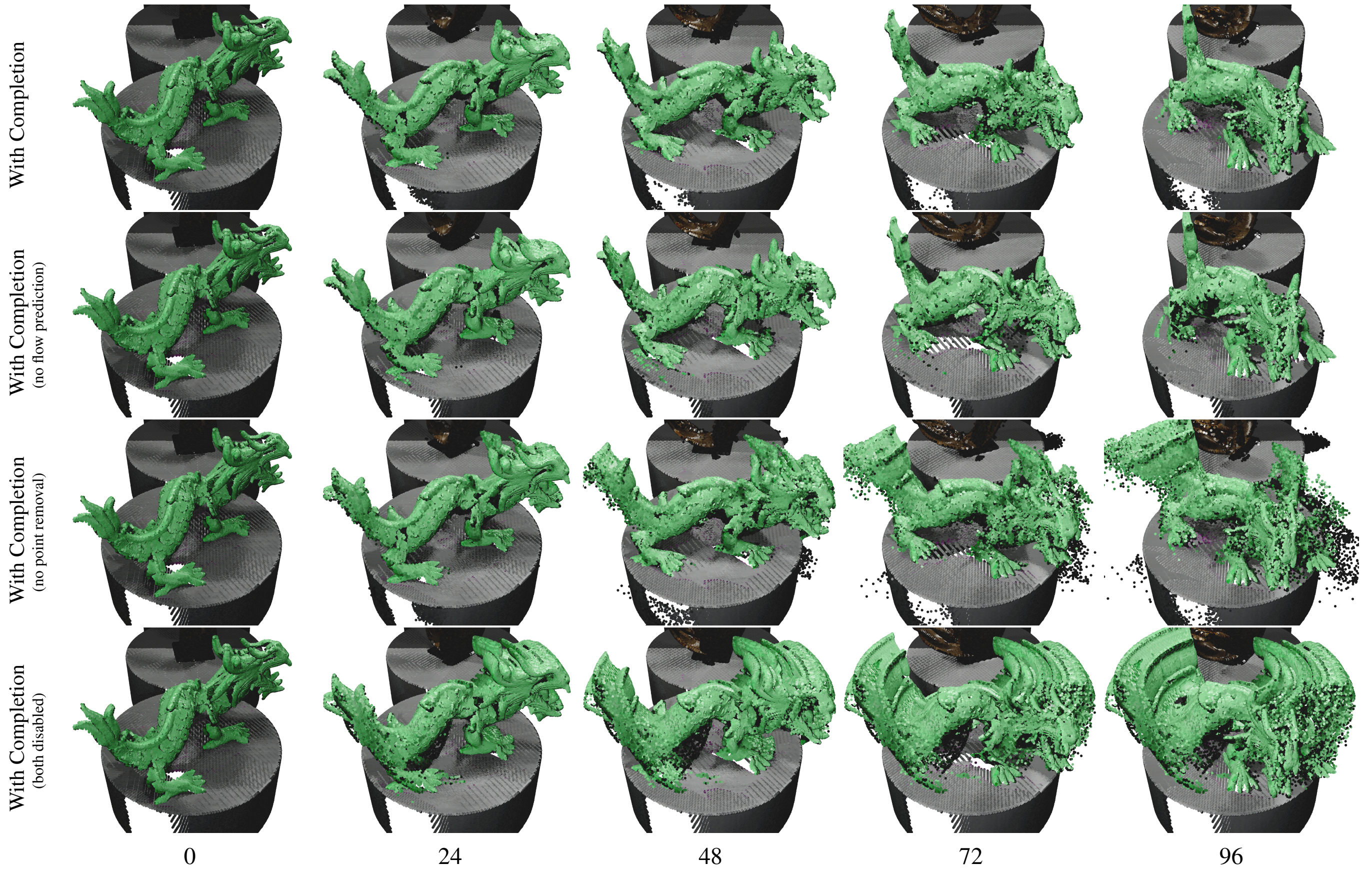


Figure 5: Visual ablation study. This figure illustrates the completion results over time when individual components of our pipeline are deactivated: Flow Prediction (second row) and our principle for removing drifting points (third row), as well as both disabled. It is evident that both components significantly influence the completion outcomes.

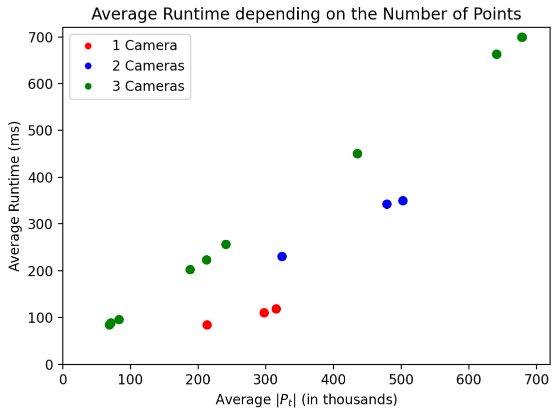


Figure 6: Visualizes the average runtime per frame for the three scenes across five configurations. It is evident that the pipeline's execution time primarily scales in $O(N \times C)$, where N denotes the number of points and C the number of cameras.

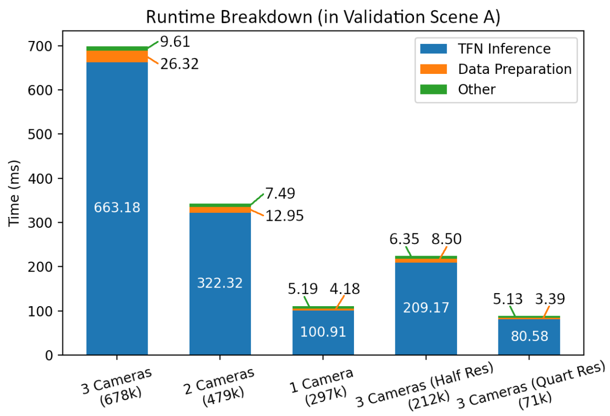


Figure 7: Displays the average runtime for different scenarios in Validation Scene A, broken down by task, with the average number of points written in brackets. It is observable that the inference of TinyFlowNet (TFN) consumes the most time and is the limiting factor for real-time performance. This is followed by data preparation—particularly the sampling including copying and coordinate system transformations of spatial and temporal positions and flow vectors, followed by the rest of the pipeline.