A. Experiment Details

A.1. Dataset Preprocessing

DrivAerNet++ provides 3D geometry in STL format. To render images, we imported each object into Blender and used the BlenderNeRF [47] extension to generate views in the same format as the Blender dataset. We set the axis-aligned bounding box to 4 and used a camera radius of 6. The first 200 rendered images were used for training or as candidate views, and another 100 were reserved for testing. The camera order was randomized. We set the background to transparent and assigned the object a gray color with hex code `#909090`.

A.2. Simulation Settings

Fluid Initialization. For each experiment we select a flow direction $d \in \mathcal{S}_d = \{\mathbf{e}_x, -\mathbf{e}_x, \mathbf{e}_z, -\mathbf{e}_z, -\mathbf{e}_y\}$. Given the rigid body's bounding box $[\mathbf{r}_{\min}, \mathbf{r}_{\max}]$, an automatic fluid slab $[\mathbf{f}_{\min}, \mathbf{f}_{\max}]$ of thickness `thick` is placed adjacent to the rigid along d, with padding to avoid intersection and spanning the rigid in the orthogonal directions. The global domain $[\mathbf{d}_{\min}, \mathbf{d}_{\max}]$ is enlarged accordingly and discretized with particle spacing equal to `particle diameter`. All particles inside $[\mathbf{f}_{\min}, \mathbf{f}_{\max}]$ are initialized as fluid particles with uniform velocity

$$\mathbf{v}_i(0) = -\lambda\, d, \qquad \lambda = \texttt{vel},$$

while rigid particles remain static.

From 3D Gaussians to Simulation Voxels. To bridge neural reconstruction and physics-based simulation, we convert the current 3D Gaussian representation $\mathcal{G}_t$ into a set of volumetric boundary samples that can be consumed by our SPH simulator for fluid–solid interaction handling. We compute a global axis-aligned bounding box (AABB) from all Gaussian centers and apply additional padding proportional to the object extent to avoid boundary artifacts. Instead of explicitly discretizing a dense voxel grid, we control the spatial resolution implicitly using an octree hierarchy.

Volumetric Sampling. We use the volumetric sampling operator provided by Kaolin [18] to directly sample points inside the union of anisotropic Gaussian volumes. Given Gaussian centers, scales, rotations (converted to quaternions), and opacities, the sampler adaptively generates volumetric samples using an octree of fixed depth. Samples whose opacity falls below a threshold (0.3 in our implementation) are discarded. This produces a dense, geometry-aware point set that approximates the occupied volume of the reconstructed object. The resulting point set serves as a voxel-like occupancy representation of the solid (Section 3.2).

Rigid–Fluid Divergence Rendering. Internal particles of rigid body do not interact with fluid particles and thus always have zero divergence. Thus, when we render the divergence of a view, we only need to consider the visible surface, without any volume rendering.

A.3. Training Schedule

To reflect practical scenarios where view acquisition is costly, we intentionally use biased initial views that provide only partial geometric coverage. Following FisherRF [28], let n_{selected} be the number of views already chosen at the current iteration. The model is trained for $n_{\text{selected}} \times 100$ iterations before selecting the next view from the candidate pool with the NBV algorithm. After the full view budget is used, the model continues training until it reaches 30k iterations.

During view selection, we follow the same overall view budget as the baseline NBV methods. Let B denote the total view budget. At each step, we set the top-K size to the remaining budget, i.e.,

$$K = \max\{B - n_{\text{selected}}, 2\}.$$

Thus, the number of candidates K considered by FluidGaussian decreases as more views are selected.

B. More Results

We provide additional quantitative results to further evaluate the effectiveness, generality, geometric accuracy, and efficiency of FluidGaussian.

B.1. Random View Selection Ablation

To isolate the contribution of our physics-aware NBV strategy, we compare random view selection and FluidGaussian under the same reconstruction baselines. As shown in Table 6, replacing random selection with FluidGaussian consistently improves PSNR across Blender, Mip-NeRF360, and DrivAerNet++ for both ActiveNeRF and FisherRF.

Table 6. Comparison between random view selection and FluidGaussian under the same reconstruction baseline. PSNR is reported.

Method	Blender	Mip	DrivAer
ActiveNeRF + Random	21.547	12.415	18.770
ActiveNeRF + FluidGaussian	**21.764**	**12.479**	**18.873**
FisherRF + Random	23.631	15.326	20.022
FisherRF + FluidGaussian	**24.743**	**15.550**	**21.073**

B.2. Results on ShapeNet

We additionally evaluate FluidGaussian on 7 ShapeNet objects. As shown in Table 7, FluidGaussian improves PSNR and SSIM, and reduces LPIPS in both cases.

Table 7. Comparison between the baseline and FluidGaussian on 7 ShapeNet objects.

Method	PSNR↑	SSIM↑	LPIPS↓
ActiveNeRF	24.828	**0.961**	0.064
ActiveNeRF + FluidGaussian	**25.653**	0.960	**0.063**
FisherRF	27.604	0.974	0.041
FisherRF + FluidGaussian	**27.807**	**0.976**	**0.034**

B.3. Chamfer Distance

We sample point clouds from the reconstructed Gaussians and from the corresponding perfect geometry (defined in Section 4.3), and compute the Chamfer Distance between them. As shown in Table 8, FluidGaussian achieves lower Chamfer Distance than both ActiveNeRF and FisherRF baselines, indicating improved geometric accuracy.

Table 8. Chamfer Distance comparison on the Blender dataset. Lower is better.

Method	Chamfer Distance ↓
ActiveNeRF	0.0103
ActiveNeRF + FluidGaussian	**0.0055**
FisherRF	0.0165
FisherRF + FluidGaussian	**0.0046**

B.4. Time Cost

We also compare the running cost of FluidGaussian and the baselines. Each fluid simulation takes 1.23 minutes on average. The total cost per geometry is about 45 minutes for FluidGaussian, compared with about 17 minutes for FisherRF and 18 minutes for ActiveNeRF. Therefore, FluidGaussian is only about 2.5× overall. At approximately 64^3 resolution, our simulator runs at 0.098 seconds per step, which is comparable to NeuralFluid [33], which reports 0.105 seconds per step.

C. Additional Related Works

Due to page limits, we defer our complete related works to this section.

C.1. Physics-Informed 3D Reconstruction

Recent advances in physics-aware reconstruction incorporate physical priors into neural representations to improve geometric accuracy and physical plausibility. Recent works [9, 34] integrate physically grounded appearance constraints, such as reflectance modeling and illumination-consistent augmentations, to improve visual quality. Other works enforce physically plausible behavior of geometry by introducing physical simulation constraints, including continuum-material modeling, particle-based dynamics and differentiable Navier–Stokes solvers for fluid–structure [32, 33, 43]. Both of these directions enhance reconstruction quality and encourage physical consistency, but they do not consider physical interaction during the reconstruction process, and therefore they cannot shape the reconstructed geometry according to functional importance. In contrast, our method introduces physics-based metrics to prioritize the reconstruction of functionally important regions.

C.2. Active Reconstruction

Active view reconstruction aims to improve 3D scene recovery by adaptively selecting informative camera poses. Building on the success of radiance-field representations [30, 40], recent approaches [28, 46] employ uncertainty estimation and Fisher information to guide NBV selection, effectively reducing redundancy and improving visual reconstruction quality. However, these methods remain purely *visual* in nature: their objectives optimize photometric fidelity, but they neglect the underlying physical plausibility of reconstructed geometry. In contrast, our work introduces physics-aware active reconstruction, where view selection is informed not only by visual uncertainty but also by simulation-derived physical metrics.

C.3. Functional Intelligence in 3D Reconstruction and Generation

Beyond purely physics-informed priors, a complementary line of work studies 3D representations that are organized around what actions they afford. At the object level, datasets such as 3D AffordanceNet and PartNet-Mobility annotate shapes with dense affordance labels and articulation, enabling per-point functional fields instead of only class or part tags [14, 51]. At the scene level, SceneFun3D and functional scene reconstruction frameworks explicitly model interactive elements, motion parameters, and actionability so that the reconstructed digital twin can be operated in simulation [13, 20]. In graphics, ICON and follow-up work on neural functionality and functionality-focused surveys argue for descriptors built around object–object interactions and functional similarity rather than purely visual similarity [24–26]. Inspired by this notion of functional intelligence, our FluidGaussian approach treats simulation-derived fluid responses as functional signals and biases active reconstruction toward regions most critical for downstream fluid–object interaction.

C.4. Uncertainty in radiance fields.

Recent work has developed explicit uncertainty estimators for radiance fields, including stochastic or variational formulations such as ActiveNeRF [46], Bayes' Rays [19], and FisherRF [28], and more recent manifold-based sampling methods [38]. These methods quantify epistemic ambiguity in the radiance field by sampling or approximating distributions over appearance and geometry parameters. While all methods have shown strong performance

in image-space tasks, these uncertainty estimates remain visual-driven: they measure how well the current radiance field explains the input views or the distribution of plausible scene attributes. They do not capture whether the reconstructed geometry behaves reliably under physical interaction. In contrast, our FluidGaussian approach targets a different notion of reliability: simulation-propagated physical quality, which cannot be inferred from photometric evidence alone. Thus, our divergence signal complements, rather than replaces, existing uncertainty formulations.