

POp-GS: Next Best View in 3D-Gaussian Splatting with P-Optimality

Supplementary Material

Table 8. Results on Single View Selection with 20 Views on the Blender Dataset.

Method	PSNR (\uparrow)	SSIM (\uparrow)	LPIPS (\downarrow)
Uniform Sampling	26.15	0.918	0.084
FisherRF	27.12	0.925	0.079
A-Opt. (Simple)	24.88	0.908	0.094
E-Opt. (Simple)	24.87	0.901	0.097
T-Opt. (Simple)	27.29	0.929	0.076
D-Opt. (Simple)	27.25	0.930	0.075
D-Opt. (Block)	<u>27.28</u>	0.930	0.075

Table 9. Results on Batch View Selection on Blender dataset.

Method	PSNR (\uparrow)	SSIM (\uparrow)	LPIPS (\downarrow)
Uniform Sampling	26.64	0.925	0.074
FisherRF	27.64	0.932	0.069
A-Opt. (Simple)	25.61	0.916	0.082
E-Opt. (Simple)	24.99	0.908	0.087
T-Opt. (Simple)	27.89	<u>0.936</u>	<u>0.065</u>
D-Opt. (Simple)	<u>27.87</u>	0.937	0.064
D-Opt. (Block)	27.80	0.935	<u>0.065</u>

6. Twenty View Blender Results

Due to page constraints, we were unable to report the results of each method on twenty view selection of the Blender dataset with the single or batch view schemes. In the main paper, we focused on ten views with the Blender dataset, as we find that the results saturate when more views are added.

Single view selection results are shown in Table 8, where FisherRF, T and D optimality achieve significantly higher performance than uniform sampling. T and D optimality outperform FisherRF, and have similar saturated results to each other.

Results on the batch view selection are demonstrated in Table 9, where we find similar results as the twenty view experiment. FisherRF, T and D optimality outperform the other baselines with T and D optimality achieving the highest results. Due to the large number of views and simple scenes, results are saturated between T and D optimality.

7. Keyframe Selection

Next we present results from the keyframe selection experiment on the Mip-Nerf360 dataset. In this experiment, we find a large performance gap between FisherRF, A and E Optimality and the other methods similar to with the

Table 10. Results on Keyframe Selection on Mip-Nerf360 Dataset.

Method	PSNR (\uparrow)	SSIM (\uparrow)	LPIPS (\downarrow)
Uniform Sampling	18.30	0.560	0.435
FisherRF	15.66	0.471	0.515
A-Opt. (Simple)	15.67	0.479	0.519
E-Opt. (Simple)	15.96	0.475	0.524
T-Opt. (Simple)	<u>18.66</u>	<u>0.560</u>	<u>0.425</u>
D-Opt. (Simple)	18.57	0.559	0.426
D-Opt. (Block)	18.73	0.571	0.417

Blender dataset. T and D optimality outperform the uniform sampling baseline, however results are saturated with ten well-chosen views. Despite the performance saturation, the block diagonal approximation leads to a noticeable improvement in SSIM and LPIPS metrics.

8. Render Quality Correlation on All Scenes

In this section, we first provide a more detailed explanation of the experimental setup for our study of the correlation between information gain and render quality. Next, we provide the sparsification plots for the remaining objects in Fig. 6.

In addition to identifying the most important training images, another key problem for applying 3D-GS to real-world applications is uncertainty quantification. Since information gain is dependent on the amount of information already present in a trained 3D-GS model at a candidate view, we would expect an inverse relationship between information gain and render quality. Therefore, we leverage sparsification plots to study the correlation between information gain and render quality. Intuitively, if a 3D-GS model has already observed data similar to a view, the method should quantify small information gain and the render should have a high reconstruction quality. Similarly, a candidate view with high information gain implies the model has limited information on the viewpoint, and the render would likely have poor reconstruction quality. This also follows from the inverse relationship between uncertainty and information stated in Section 3.2.1.

In order to study this relationship, we leverage sparsification plots. The primary idea behind sparsification plots is to sort candidate views by information gain, and observe the relationship between information gain at candidate views and render quality at the candidate views. In this experiment, we train a single 3D-GS model on ten randomly chosen views so that there is new information at the remaining

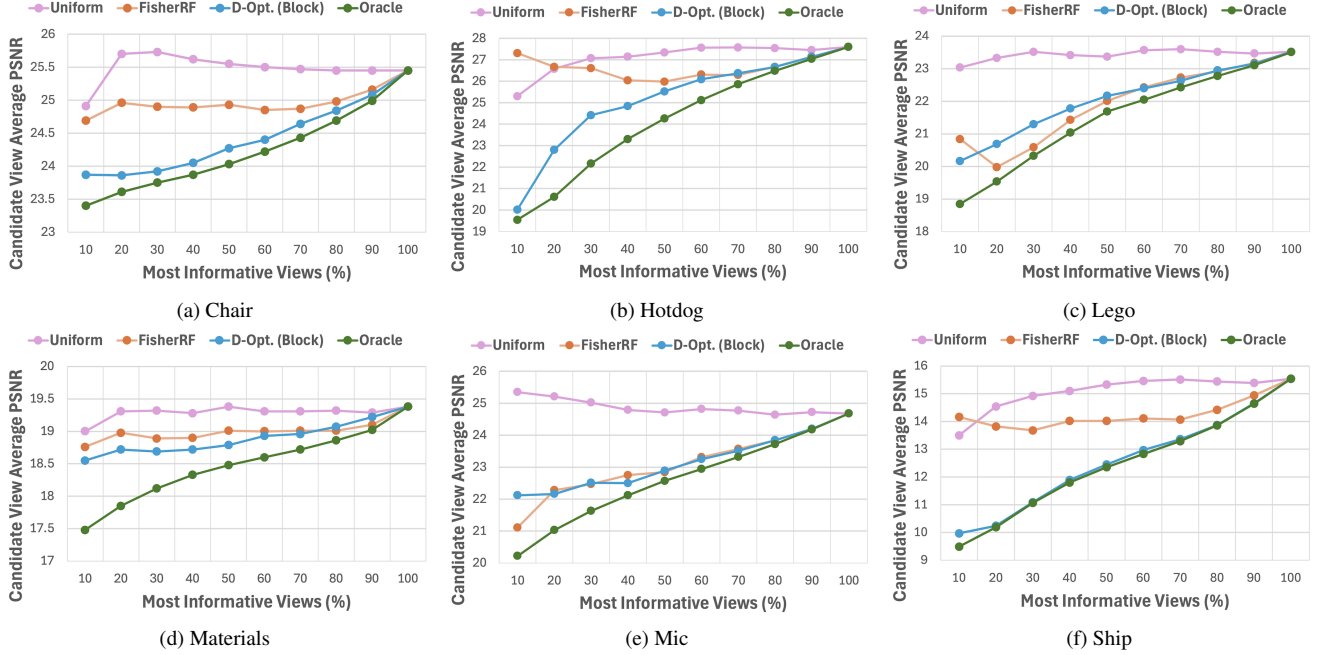


Figure 6. Uncertainty correlation plots on all remaining scenes of the Blender dataset. The oracle represents a perfect sorting of the views by PSNR. If the information gained by candidate views is well calibrated, the ordering should be similar to that of the oracle, resulting in a low value at the left of the plot which contains the average reconstruction quality of the most informative views.

views in the dataset. Next, each method sorts the remaining views by estimated information gain from most information to least information. The purpose of the sparsification plot is then to study how the views are sorted by estimated information gain.

To create the sparsification plot, the sorted views are organized into groups based on decile. For example, for one hundred candidate views the first group would contain the ten most informative views and the final group would contain the ten least informative views. Next, the groups are combined iteratively beginning from the most informative views and the average PSNR is calculated for the combined groups. Therefore, the plot represents the *average* PSNR of the $x\%$ most informative views. As a result, we would expect to see a low PSNR for the most informative views at the left of the plot, and all methods converge at the right of the plot when calculating the average PSNR over all images.

For baselines we use the uniform sampling method, which should demonstrate no correlation between expected information gain and average PSNR. We also introduce an oracle baseline which directly observes the PSNR of each render and represents an ideal ordering. We compare FisherRF and D-Opt. (Block) with the baselines, where the best performing method is the method most similar to the oracle demonstrating a relationship between uncertainty and render quality.

Fig. 6 details the remaining plots for all objects in the

dataset. In the main text, we chose figures which highlighted the performance of FisherRF as well as D-Opt. (Block). D-Opt. (Block) generally has a monotomic behavior and performance near the oracle. However, FisherRF sometimes does not exhibit the same behavior depending on the object. We note that this plot is not the intended goal of FisherRF, however we would expect a strong correlation between information gain and reconstruction error as stated previously.