HP-GS: Human-Preference Next Best View Selection for 3D Gaussian Splatting

Matthew Strong and Aditya Dutt
Department of Computer Science
Stanford University
Stanford, CA 94205
{mastro1, asdutt}@stanford.edu

Abstract

In this work, we present a simple method for guiding the next best view in 3D Gaussian Splatting via human preferences. The next best view in 3D Gaussian Splatting enables for uncertainty-aware reconstruction in an autonomous manner; however, the selection of the optimal next best view remains fairly unexplored in the literature. Motivated by the idea that humans are naturally adept at recognizing visual uncertainty and can use common sense to understand where to view next in scenes, we propose to use a visual human preference model that distills human preferences into a simple ResNet model. Given many pairs of images from randomly trained Gaussian Splat(s), a human selects which image is most poorly reconstructed. This model is then utilized during real-time Gaussian Splatting training to greedily select the next best view. We then propose a novel view selection method of maximizing the feature space cosine distance from the *training set*, which proves to surpass a naive Condorcet winner solution. We validate our method on the Mip-NeRF 360 dataset, which consists of several challenging indoor and outdoor scenes, and is considered the standard for view selection of radiance fields. Our results prove to be on-par and in some cases *surpassing* state-of-the-art view selection methods. Our video link can found here. We release our code on Github.

1 Introduction and Related Work

In recent years, 3D Gaussian Splatting (3DGS) [6] has become a staple of visual 3D reconstruction. Only requiring images and camera poses (which can be estimated with off-the-shelf Structure from Motion algorithms such as COLMAP [11]), 3DGS has proven to be photorealistic and geometrically precise 3D representation. Further, the real-time (70+ FPS rendering speeds on a standard desktop GPU) speeds of 3DGS make it a suitable candidate over its implicit and neural counterpart NeRFs [9]. With this, Gaussian Splatting is already finding applications such as robotics [13, 14, 12, 2], virtual reality [5], and text to 3D [16], with unmatched speed and quality.

However, a well-trained and constructed Gaussian Splat generally requires hundreds of views to construct a scene, because splats trained on few views will overfit to each training view, leading to degradation of color *and* depth reconstruction on out of distribution testing views. In this case, it is desirable to train a Gaussian Splatting model on a low number of impactful views. If each view is both impactful and selected in an intelligent manner, practitioners in the field will not have the mental burden of selecting many good views.

In general, robust splats in the wild can be optimized by **a).** enhancing the impact of each view ([3, 13, 17, 10, 7]) and/or **b).** selecting specific views that improve the reconstruction ([13, 4, 8]). Works increasing the impact of each view often incorporate priors to improve the generalization of

Submitted to CS 329H at Stanford.

Gaussian Splat, frequently relying on monocular depth maps to seed and initialize a scene. These methods are often improved by large vision models that add data-driven improvements to Gaussian Splatting, which is traditionally considered a non-deep learning method. However, the problem of next best view selection in **b).** still relies on analytical solutions that claim mathematical optimality, but in practice reduce to fairly naive assumptions that humans would never make. In fact, [13] found that random view selection as compared to state-of-the-art is still a challenging baseline to beat, as random views can escape local minima that other methods suffer from.

However, at present, the best next view selection algorithms are still built on these weaker assumptions. All three works ([13, 4, 8]) in next best view selection still rely on theory that stems from slightly limited underlying assumptions – for example, FisherRF [4], considered SOTA in view selection for Gaussian Splatting, operates on the fact that optimal next best views will have a large change in color. This assumption breaks down in color-rich environments that humans naturally navigate in. [13] improved upon this by guiding new views based on depth in robotic environments, as most scenes in the wild are geometrically smooth. While this is a stronger assumption, it lacks the common sense that humans would use when selecting a new view. [15] models some uncertainty in Gaussian Splatting, but it is limited to a discrete set of classes and is not open world, which requires a model that is trained on an extensive amount of classes and is not easily applicable for open-world active view selection.

This leads to the question, what might be some effective ways to autonomously select new views in a Gaussian Splat in a common sense manner? A human can clearly tell when a view in Gaussian Splatting is uncertain and poorly constructed, but a human-in-the-loop violates the condition for autonomous view selection. We propose to *learn human preferences* by distilling human preferences of views into a simple vision model. This way, we can encode human preferences in a model, and simply call this lightweight model (in our case, simply a Resnet34 model) during view selection once per view, or in a batch.

Concretely, our method, which we call **HP-GS**, or Human Preference-based Gaussian Splatting, is a framework for using human preferences for active view selection in Gaussian Splatting. First, we render many views in a randomly trained Gaussian Splat per scene over different time intervals. Then, we collect many pairs of rendered images, where some are exactly matched with training views, and other views are extremely poorly constructed. With each pair, we have a human annotate the pairs by selecting the more uncertain one, which we find to be easier than assigning manual values of uncertainty to views.

We then learn a simple human preference model with Bradley-Terry by using a a light Resnet feature encoder, which predicts the probability of an image being more *uncertain* than others. Intuitively, our model learns to understand what humans find visually uncertain in a scene. When a Gaussian Splat queries for the next best view, we can iterate through the *candidate views* and select the view that has the largest *cosine distance* from the median feature vector of the *training views*. We find that this insight performs better than inefficiently taking the Condorcet winner of all of the candidate views.

During deployment of the model, only one forward pass is needed per image, which is not much slower than a backward pass on a few million Gaussians performed in recent state of the art selection method FisherRF [4].

We note limitations and ethical concerns of our method at the end of the paper, which we believe prevent it from being utilized out in the wild. We then propose modifications that are an important next step for guiding views in 3DGS with human preferences.

2 Preliminaries

3D Gaussian Splatting is a 3D representation that represents a scene with 3D Gaussians. Each Gaussian is represented with mean $\mu \in \mathbb{R}^3$ and covariance $\Sigma \in \mathbb{R}^{3\times3}$. The covariance can be computed as the product of a diagonal scale matrix $S \in \mathbb{R}^{3\times3}$ and rotation matrix $R \in \mathbb{R}^{3\times3}$, which outputs $\Sigma = RSS^TR^T$. Each Gaussian primitive also is parameterized with an opacity value $\alpha \in \mathbb{R}$ and spherical harmonic parameters for RGB color. The color and depth of each pixel in a *rendered* image can be computed as blending points along a single camera ray, which is defined as: $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with camera origin $\mathbf{o}$ and orientation $\mathbf{d}$. Then, color $C(\mathbf{r})$ and depth $D(\mathbf{r})$ are

computed by blending points intersecting the camera ray:

$$\hat{C} = \sum_{i \in N} c_i \alpha_i T_i, \quad \hat{D} = \sum_{i \in N} d_i \alpha_i T_i, \tag{1}$$

where $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$, which is the transmittance, concretely defined as multiplying the previous Gaussian opacity values intersecting the ray. During training, the parameters of each Gaussian are optimized with gradient descent to minimize the following photometric loss between a ground truth and rendered image: as $\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{SSIM}}$, where $\mathcal{L}_1$ is the L1 loss between the RGB values of the rendered and ground truth image, and $\mathcal{L}_{\text{SSIM}}$ is the **S**tructural **S**imilarity **I**ndex **M**easure (SSIM) loss. Further, we can extend the loss to include $\mathcal{L}_{depth}$ and other priors to prevent overfitting and to improve reconstruction. We use a depth loss in disparity space to heighten the impact of each view, but this is not the focus of this paper.

3 Method

3.1 Initial Data Collection and Model Training

To train a scene specific model, we desire to collect diverse pairs of data for a human to label. To achieve this, we first train a Gaussian Splat with several views for each scene. The idea here is that a reasonably conditioned Splat will have areas both of low color loss and high color loss with respect to ground truth, meaning that there will be areas of high and low uncertainty. Specifically, we train a Splat on each scene for 15000 steps, and every 2000 steps, we render the entire camera trajectory that was originally taken and collect every RGB image.

Once all of the images are collected, we randomly shuffle the dataset $\mathcal{D}$ of scene s, and take n random pairs of images for our scene specific dataset $\mathcal{D}_s$. While this dataset makes the false assumption that views at *different timesteps* can be compared, we find that this augmentation adds extra diversity for the model to learn. The labeled dataset is then used to train a simple ResNet-34 model to correctly determine the candidate next view, which is the more uncertain image given a pair of images. This equation is briefly shown below.

$$\begin{aligned} f_1 &= F(x_1) \\ f_2 &= F(x_2) \\ f_1 &= l(f_1) \\ f_2 &= l(f_2) \\ p &= \frac{\exp(x_1)}{\exp(x_1) + \exp(x_2)} \end{aligned} \tag{2}$$

Here, we feed image x_1 and x_2 into Resnet feature extractor F, receiving output features f_1 and f_2. We pass these features through a linear layer l and compute the probability with a simple preference-based Bradley-Terry model. Figure 2 provides a visualization of the model architecture. The probability is whether image x_1 is preferred over x_2. An example of two images the labeler is asked to review is shown in Fig. 1. Visually, the first image contains many spurious Gaussian artifacts and is not as well-constructed as the second image, hence the first one is selected. While an LLM such as ChatGPT could assist in easier examples such as this one, it would not be as reliable for harder pairs of views that require a human annotator over 5 seconds to decide on the better view. However, we believe it to an interesting potential pretraining step where a human can perform *corrective* actions if the initial prediction is wrong. Other outputs such as features from CLIP or DINO could also be useful for determining if a view is better – we leave this for future work in selecting views based on semantic uncertainty.

3.2 Gaussian Splatting Training

With a trained model for scene s, we can immediately deploy for real-time Gaussian Splatting training. We initialize a scene from m views. Every 2000 steps, we desire a policy to select a new view.

Figure 1: An example of two 3DGS renderings a human labeler would need to compare.

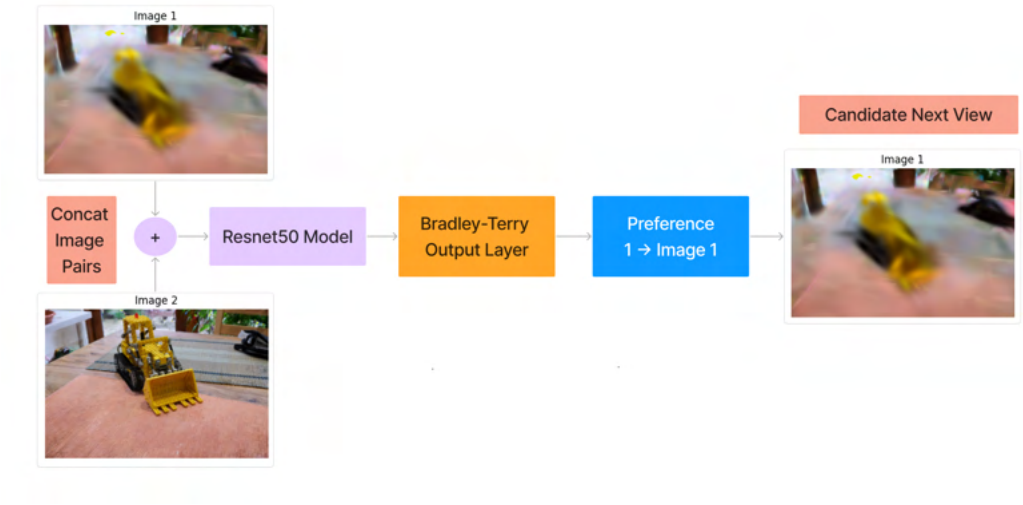

Figure 2: Visualization of Next-best view Selection using a ResNet50 model. The output from the ResNet model is passed into a Bradley-Terry reward output that chooses the candidate next view.

A first approach is to look to FisherRF [4] for next best view selection, where the problem of selecting the next best view is formulated as maximizing the Fisher information gain between candidate RGB camera poses x_i^{acq} and views y_i^{acq} and captured training RGB views D^{train}, given Gaussians' parameters w^*.

$$\begin{aligned} &\mathcal{I}[w; \{y_i^{\text{acq}}\} | \{x_i^{\text{acq}}\}, D^{\text{train}}] \\ &= H[w^* | D^{\text{train}}] - H[w^* | \{y_i^{\text{acq}}\}, \{x_i^{\text{acq}}\}, D^{\text{train}}] \end{aligned} \tag{3}$$

$H[\cdot]$ is the entropy of the Gaussian Splat. However, we abandon the notion of information gain and end up turning to developing a simple uncertainty metric, where we take the argmax of a function U

$$\arg\max_{x \in X} U(x)$$

and greedily take the best available view. The first approach we can take is Condorcet winner (shown in Fig of many duels over m candidate views. We can take a duels, and take the view which wins the

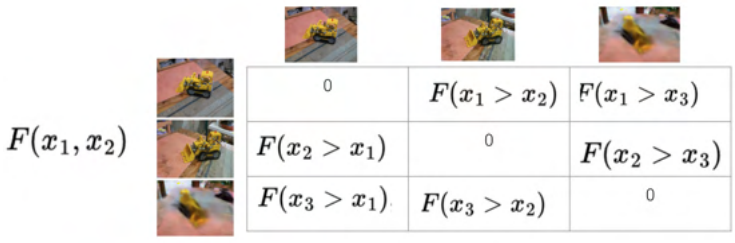

Figure 3: Taking the Condorcet winner over many candidate views, in this case three images, does not take into account the training views, and makes many (n^2) comparisons between potential images.

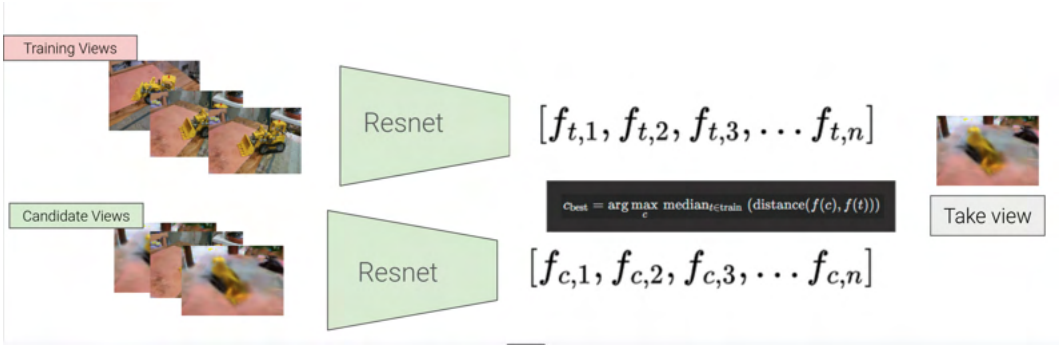

Figure 4: We propose a feature space comparison between training and testing views.

most duels over other views, returning the winner. While this seems effective, it suffers from two issues: 1). querying for the next best view is $O(n^2)$ and requires a quadratic amount of calls to our model, and 2). does not use knowledge of the *current* training set. We desire a quantitative measure of uncertainty of a candidate view in $O(n)$ time.

We propose a novel method to deal with this issue. We first run the preference model over the training views, collecting the output features from the trained Resnet. Then, we run the model once over each candidate view, and select the candidate view feature that has the **highest cosine distance** between the median feature vector of the training set. We use the median feature vector as the mean feature has a tendency to be strongly influenced by outliers. By using human-preference features, we can quantify the uncertainty between the train and candidate set. This method is $O(n)$, meeting our goal of a fast metric to use human preferences to select views. A figure of this method can be seen in Fig 4.

4 Experiment and Results

4.1 Dataset

We collect our dataset from the publicly available Mip-NeRF 360 dataset [1], with examples shown in Fig 5. In the vision community, this is considered the benchmark for view selection.

4.2 Experiment Details

It consists of 7 indoor and outdoor scenes. For data collection, we use interactive matplotlib and pandas in Python to label data. We train a simple off-the-shelf Pytorch Resnet model and use the gsplat library for our 3D Gaussian Splatting implementation. We implement active learning from scratch in the repo and open-source it at our codebase link. All experiments are run onboard a NVIDIA RTX 3080 GPU.

4.3 Evaluation of Resnet Model

We evaluate the ResNet model on test pairs of images taken from the generated dataset. Fig 12 showcases a few examples of the model evaluated on the kitchen scene. We can see that the model correctly identifies the more uncertain image pair even when there is fewer distinctive uncertainties or artifacts in the image. This highlights the benefit of providing human-labeled preferences to distinguish between ambiguous or equally-noisy images.

4.4 Mip-NeRF Dataset Evaluation

We train scene specific models (around 500 pairs of images per scene model was sufficient for our scene). We show the PSNR, SSIM, and LPIPS of random view selection, FisherRF [4], and our method (as well as the Condorcet winner ablation on one scene). We select new views every 2000 steps for every method for a fair comparison, and keep the initial training views the same. We run every scene to 15000 steps during training, and compare the resulting visual metrics.

Figure 5: Example scenes from our dataset. These scenes are multi-object, cluttered, and colorful, making them a challenging benchmark for next-best-view selection.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓
Kitchen	Random	11.10	0.18	0.67
	FisherRF	11.27	0.19	0.67
	Condorcet Winner	11.20	0.17	0.66
	HP-GS	**11.34**	**0.30**	**0.66**
Room	Random	12.39	0.39	0.58
	FisherRF	12.45	0.38	0.56
	HP-GS	**13.15**	**0.51**	**0.56**
Counter	Random	11.79	0.26	**0.61**
	FisherRF	**11.84**	0.26	**0.61**
	HP-GS	**11.84**	**0.27**	**0.61**
Bonsai	Random	10.02	0.25	0.64
	FisherRF	**10.26**	**0.27**	**0.63**
	HP-GS	9.69	0.26	0.64

Table 1: Indoor Scene Results. The best results are in **bold**.

Across the board in our indoor scene results Table 1 and outdoor scene results TAble 2, our method is competitive with state of the art FisherRF, beating it in the majority of the scenes for PSNR (PSNR is the most important metric in Gaussian Splatting). Interestingly, random view selection is still consistently competitive, showing its challenge as a baseline to beat. Qualitatively, our method selects views that explore the scene more broadly. In general, this leads to strong performance, but scenes such as the bike scene (Fig 8) and garden scene (Fig 6) do not do as well because our model explores the scene broadly but sparsely due to the low number of views added to the scene (only 7 views added). In this case, future work entails training our splats up to 30000 steps with view selection turned on to see if our method more consistently outperforms other the other view selection algorithms.

Operating in feature space is useful, as Fig 9 highlights how HP-GS improves the human preferences of a scene.

Figure 6: GARDEN SCENE.

Figure 7: STUMP SCENE: Our method selects views that *explore* the scene, driven by our model.

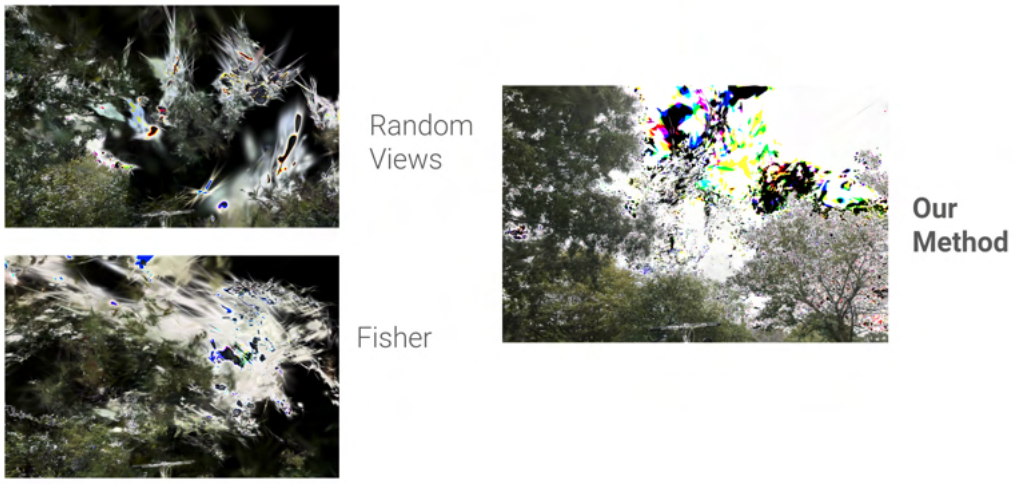

Figure 8: BIKE SCENE: While our method is not the best on the bike scene, it suggests views that explore the environment more, as seen in the sky in the above images. The method slightly suffers from exploring broadly but sparsely. Future work entails adding views to the scene to see if our method has a larger advantage.

Figure 9: Our method selects views that look more desirable to the human eye, encoded in our simple Resnet model.

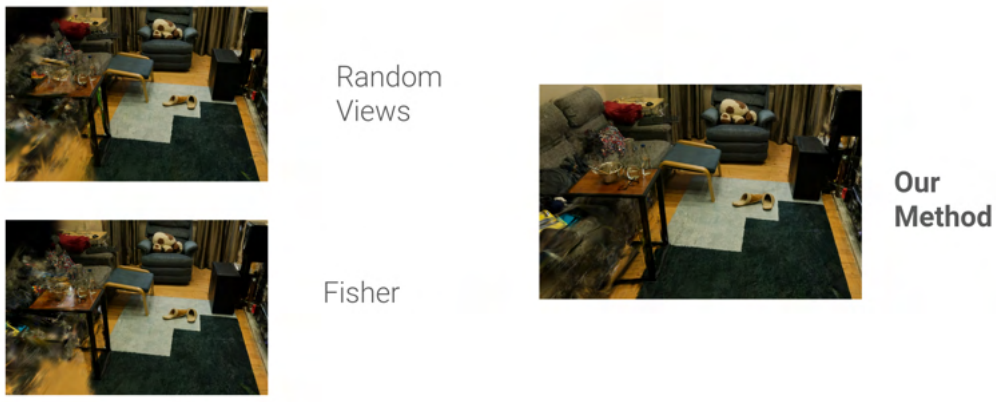

Figure 10: ROOM SCENE.

Figure 11: Our method aligns with views that a human selected, which sometimes does not improve entire scene coverage.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓
Stump	Random	15.83	0.39	0.63
	FisherRF	15.76	0.39	0.63
	HP-GS	**16.07**	**0.39**	**0.63**
Garden	Random	**11.60**	**0.19**	**0.65**
	FisherRF	11.53	**0.19**	**0.65**
	HP-GS	11.33	**0.19**	**0.65**
Bicycle	Random	**13.95**	**0.26**	**0.60**
	FisherRF	13.73	0.25	**0.60**
	HP-GS	13.89	**0.26**	**0.60**

Table 2: Outdoor Scene Results. The best results are in **bold**.

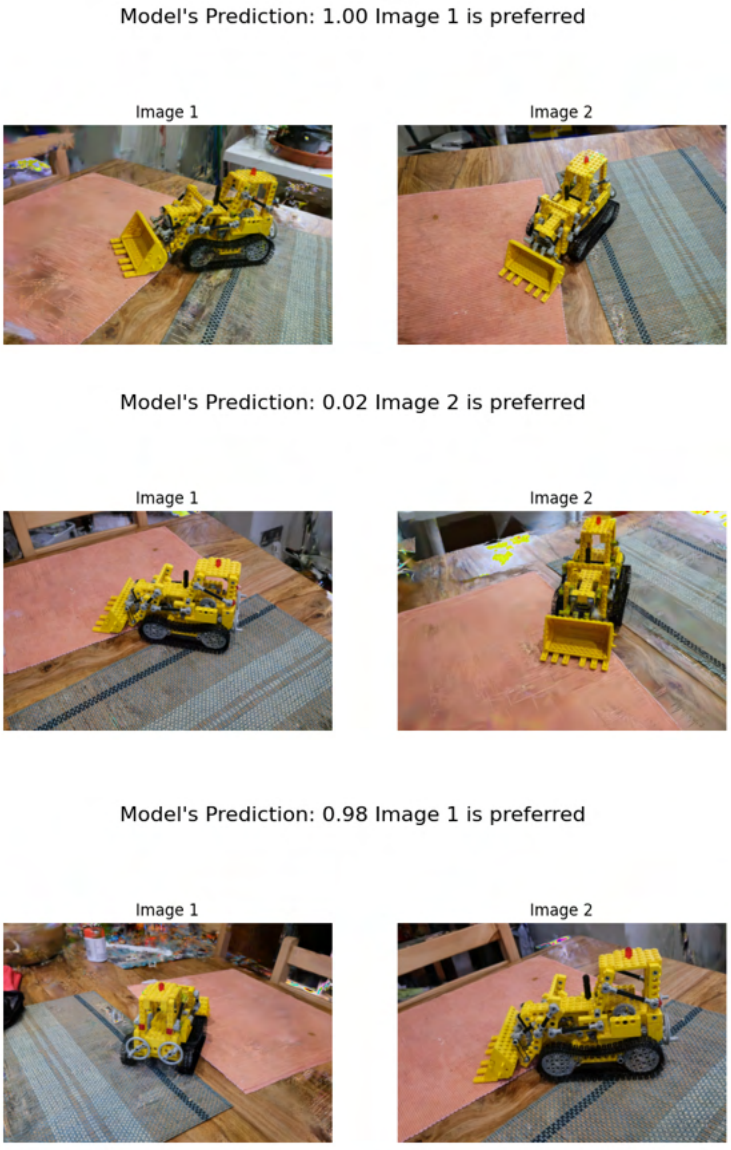

Figure 12: Evaluation of ResNet model on test image pairs from kitchen scene.

5 Conclusion and Future Work

In this work, we presented HP-GS, the first-ever method for leveraging human preferences for view selection in Gaussian Splatting. First results demonstrate the usefulness of distilling preferences into view selection for radiance fields, as HP-GS performs superior to state-of-the-art work in view selection.

The biggest limitation of our method is that we learn a scene specific model, requiring training for each scene. While this is a promising method, it is computationally infeasible in the wild, where the scenes could be theoretically infinite in amount. To address this, we will learn a large human preference model on *many* Gaussian Splats by taking many splats in the wild and learning one model to predict low quality renderings in Gaussian Splats. This way, with a robust model, it can simply be used out of the box for future scenes that demand next-best-view selection.

We also plan to perform a pretraining step with ChatGPT, which has a degree of common sense encoded within it and could recognize ill-conditioned views. Finally, we plan to ablate on the number of views added more rigorously to confirm the effectiveness of our method.

6 Ethics Statement

We list potential ethical risks of this work and how they would be mitigated.

1. **Risk 1: Biased Data Collection by Human Data Collectors**
   Description: People who are labelling data might be visually biased to certain parts of a scene and look at scenes fairly, leadings to degrading performance of view selection that could negatively affect people who want to use it in the wild.
   Mitigation: Perform a rigorous screening of people who would like to collect data by verifying their performance on a few toy scenes. Then, running bias detector models to see if views are preferred unfairly
   Research Design: The authors labelled the data in this paper. If future data collectors are added, we will screen and look at trends in their data as a critical quality check
2. **Risk 2: Biased Scene Selection**
   Description: Scene selection could be bias – even with perfect human labellers, the model may be trained on a biased dataset and perform poorly on out-of-distribution tasks that could be safety critical.
   Mitigation: Ensure each data well encapsulates the distribution of scenes Gaussian Splats may be trained on.
   Research Design: Implement a crowd-sourced (e.g., Amazon Mechanical Turk) method to collect the most common scenes people interact with (kitchens, bathrooms, workplaces, etc) and use that to inform the data collection for training the preference model.
3. **Risk 3: Misuse with Physical Systems**
   Description: Bad people could implement this framework on a robot or drone, and use it to actively explore an environment and hunt down people.
   Mitigation: Ensure that this research is not used harmfully with advocation.
   Research Design: Limit the scenes to normal street scenes that are out of scope of an aerial drone.
4. **Risk 4: Loss of Trust Due to No Explainability**
   Description: The proposed preference model currently has limited explainability. The results in the scenes are good but we did not dive deep into the nature of the model itself.
   Mitigation: Since the model is fairly small, we can visualize the feature outputs at each and every single layer of the model to understand what concretely is being learned.
   Research Design: Implement feature visualizations of different outputs and what activates to see what the model is learning instead of blindly trusting it based on experimental results.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[2] Timothy Chen, Aiden Swann, Javier Yu, Ola Shorinwa, Riku Murai, Monroe Kennedy III, and Mac Schwager. Safer-splat: A control barrier function for safe navigation with online gaussian splatting maps. *arXiv preprint arXiv:2409.09868*, 2024.

[3] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 811–820, 2024.

[4] Wen Jiang, Boshu Lei, and Kostas Daniilidis. Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information. *arXiv preprint arXiv:2311.17874*, 2023.

[5] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–1, 2024.

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[7] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024.

[8] Linjie Lyu, Ayush Tewari, Marc Habermann, Shunsuke Saito, Michael Zollhöfer, Thomas Leimküehler, and Christian Theobalt. Manifold sampling for differentiable uncertainty in radiance fields. In *SIGGRAPH Asia Conference Proceedings*, 2024. doi: 10.1145/3680528.3687655.

[9] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[10] Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. In *European Conference on Computer Vision*, pages 19–37. Springer, 2025.

[11] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] Ola Shorinwa, Johnathan Tucker, Aliyah Smith, Aiden Swann, Timothy Chen, Roya Firoozi, Monroe Kennedy III, and Mac Schwager. Splat-mover: Multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting. *arXiv preprint arXiv:2405.04378*, 2024.

[13] Matthew Strong, Boshu Lei, Aiden Swann, Wen Jiang, Kostas Daniilidis, and Monroe Kennedy III. Next best sense: Guiding vision and touch with fisherrf for 3d gaussian splatting. *arXiv preprint arXiv:2410.04680*, 2024.

[14] Aiden Swann, Matthew Strong, Won Kyung Do, Gadiel Sznaier Camps, Mac Schwager, and Monroe Kennedy III. Touch-gs: Visual-tactile supervised 3d gaussian splatting. *arXiv preprint arXiv:2403.09875*, 2024.

[15] Joey Wilson, Marcelino Almeida, Min Sun, Sachit Mahajan, Maani Ghaffari, Parker Ewen, Omid Ghasemalizadeh, Cheng-Hao Kuo, and Arnie Sen. Modeling uncertainty in 3d gaussian splatting through continuous semantic splatting. *arXiv preprint arXiv:2411.02547*, 2024.

[16] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*, 2023.

[17] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pages 145–163. Springer, 2025.

A Appendix / supplemental material

Optionally include supplemental material (complete proofs, additional experiments and plots) in appendix. All such materials **SHOULD be included in the main submission.**