

電機資訊學院 2026 實作專題競賽 BRAIN PLUS HAND

VIS4GS: Visual Analytic Tool for 3DGS Reconstruction

隊伍編號: EECS09 組長: 林楷源 隊員: 黃峰

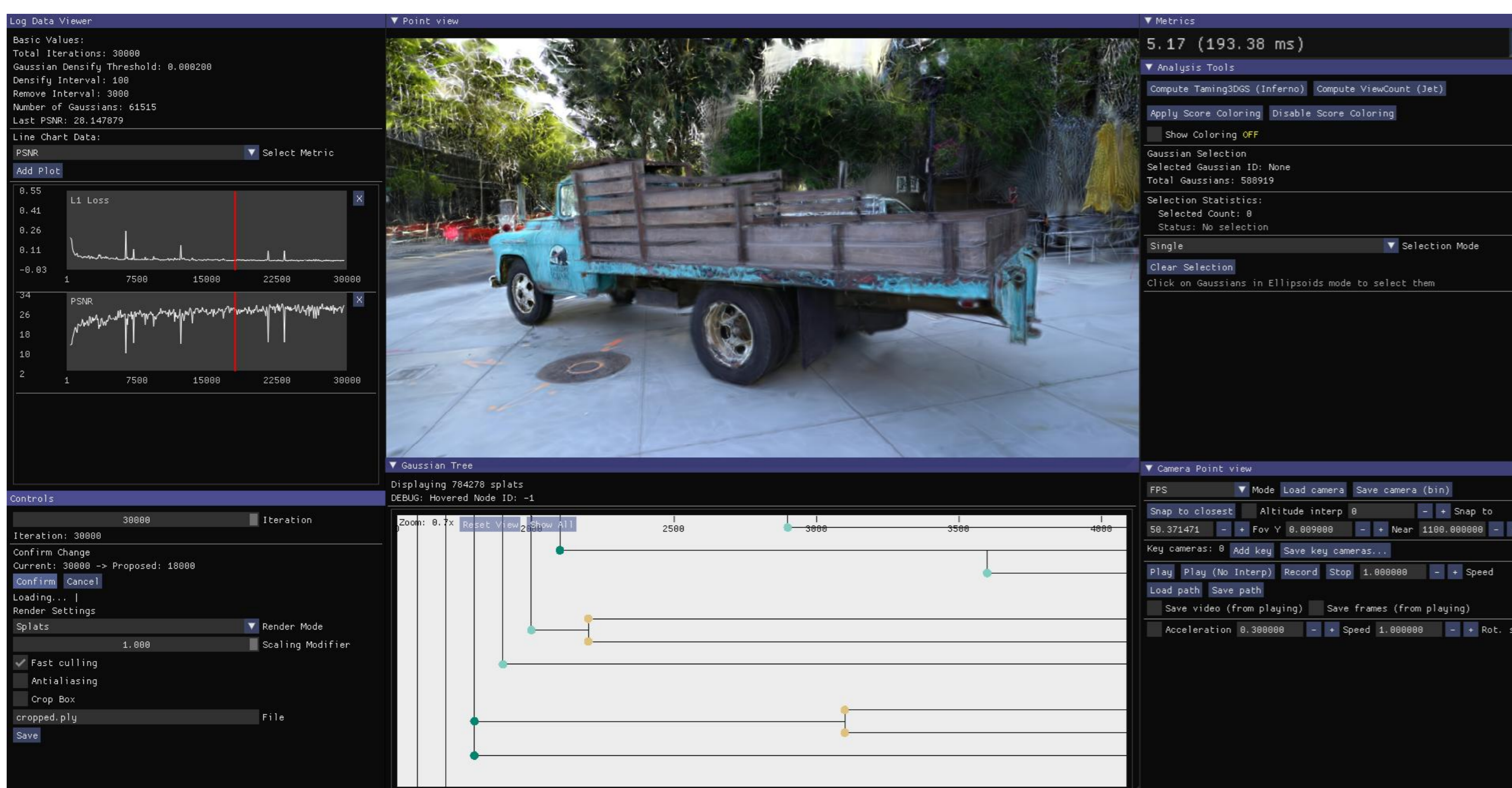


Fig. 1. User Interface for Vis4GS, based on original 3DGS viewer [1]. Components include Log Data Viewer, 3DGS Scene, Quality Score Controls, Iteration and Scene Controls, Gaussian Tree View, and Camera Controls.

Methodology

System Overview

The UI integrates all components, providing an in-depth look at 3DGS synthesis and training. 3DGS Backend exports data into different visualization tools, which are integrated into the viewer, while the quality assessment module directly interacts with scene model. A feature to change current iteration is also included.

1. Training Data Visualization

We present key training metrics (e.g., PSNR, Loss, Gaussian numbers) via numbers and line charts over iterations, exported from training using TensorBoard.

2. Gaussian Tree View

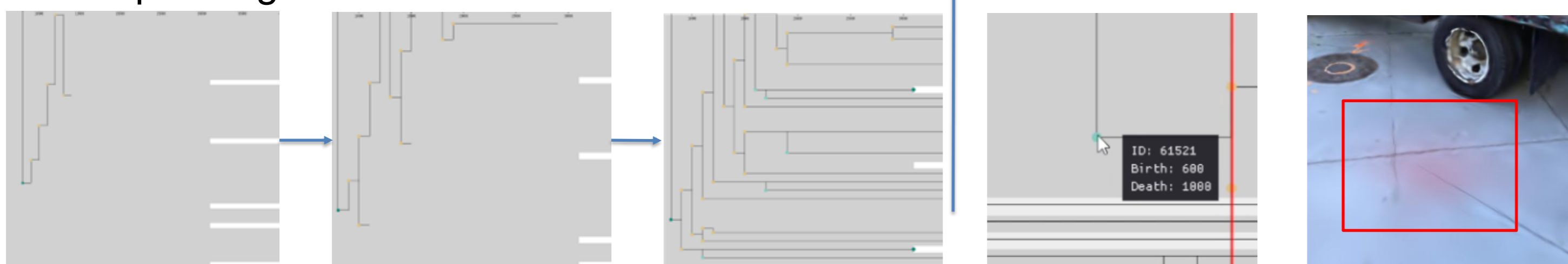
This modified tree structure visualizes Gaussian densification events showing relationships and lifespans. An LoD system, based on perception value (pV) of each node and a threshold value based on normalization on zoom, manages clutter. Hovering over a tree node highlights its corresponding Gaussian in the 3D scene.

$$pV(i) = (\alpha \cdot Score_{desc}(i)) + ((1 - \alpha) \cdot Score_{life}(i))$$

$$Score_{desc} = \frac{Descendants(i)}{\max(Descendants_{family})}$$

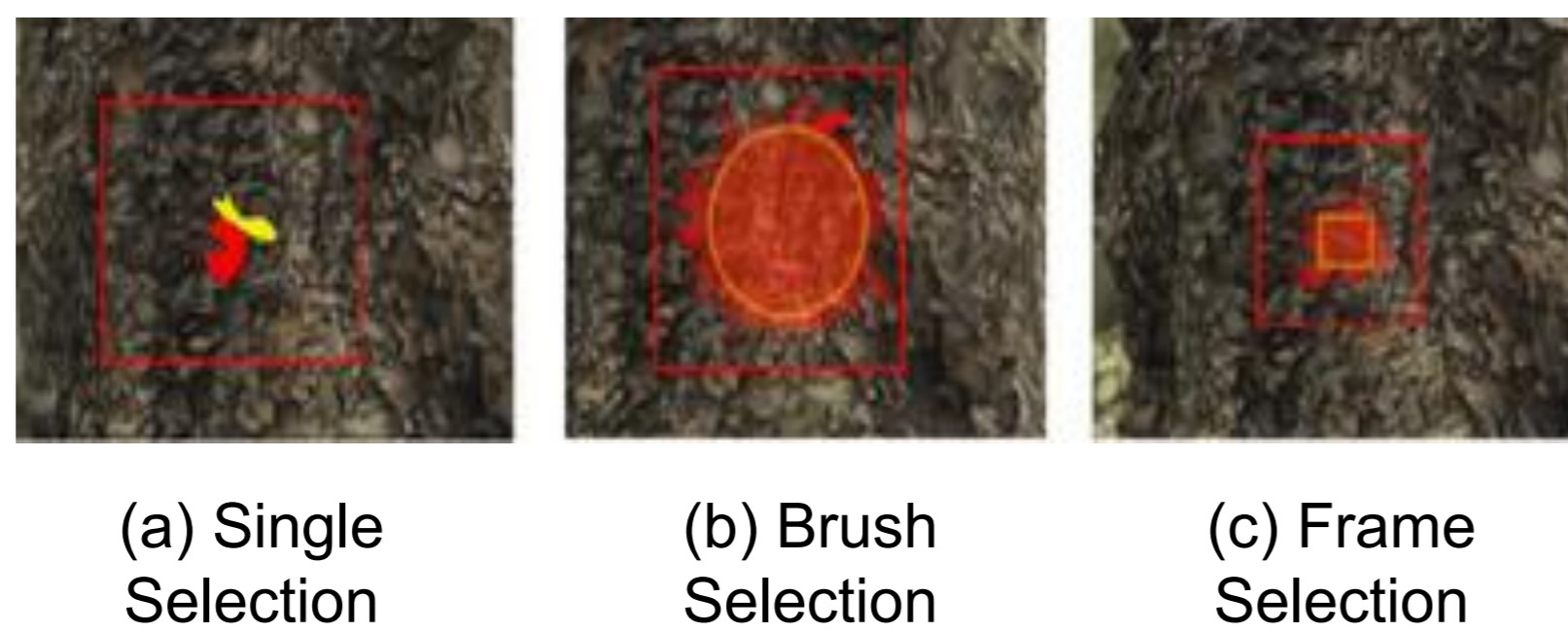
$$Score_{life} = \frac{Lifespan(i)}{\max(Lifespan_{family})}$$

$$pV_{threshold} = (1 - \frac{z - z_{min}}{z_{max} - z_{min}})^k$$



3. Gaussian Selection System

The system provides four selection modes (Single, Ray, Brush, and Frame) to isolate and inspect Gaussians in dense regions. The implementation leverages GPU rendering and octree-based ray intersection, enabling fast, real-time selection and highlighting.



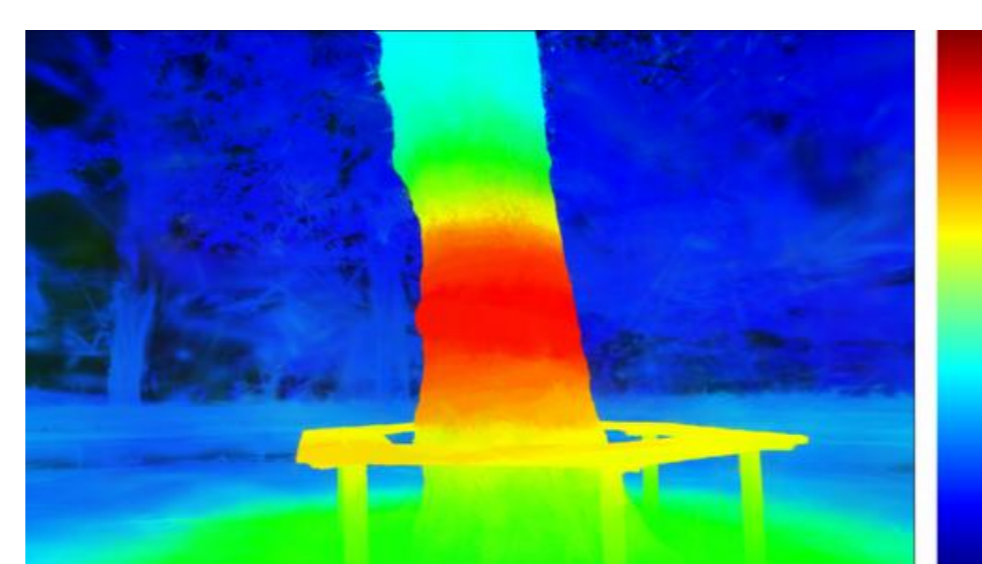
4. Quality Scoring

The per-Gaussian importance score combines geometric and photometric metrics from the current camera, following Taming3DGS [2]. The resulting values are normalized and visualized using an Inferno colormap, where brighter regions indicate higher importance and darker regions indicate low importance.



5. View Count

View Count computes how many cameras can see each Gaussian to represent dataset coverage. The normalized scores are then visualized with a Jet colormap, where red colors indicate regions observed by more cameras and blue colors are less observed.



References

- [1] Kerbl, B., Kopanas, G., Leimkuehler, T., and Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. 42, 4 (July 2023).
- [2] Mallick, S. S., Goel, R., Kerbl, B., Carrasco, F. V., Steinberger, M., and De la Torre, F. Taming 3dgs: High-quality radiance fields with limited resources. In Proceedings of SIGGRAPH Asia 2024 Conference Papers (Tokyo, Japan, Dec. 2024), SA '24, Association for Computing Machinery.

Introduction

3D Gaussian Splatting (3DGS) is a novel method enabling high-quality, real-time rendering. Despite its success, the 3DGS representation is a massive, unstructured collection of primitives. This "black box" nature makes it difficult for users to understand *why* visual artifacts (e.g., "floaters," blurry patches) occur, which hinders debugging and scene optimization. To address this, we introduce a visual analytics tool that provides a detailed, understandable view of the 3DGS synthesis process. Our goal is to empower users to pinpoint the cause of artifacts and determine the importance of each splat. Our tool integrates several components to analyze the scene, including:

Training & Densification Analysis: A Gaussian genealogy tree and iteration view to visualize the relationship between splats and how the scene evolves during training.

Splat Inspection System: A selection system, per-Gaussian quality scoring, and a training-camera view-count analysis to isolate and inspect problematic or salient Gaussians.

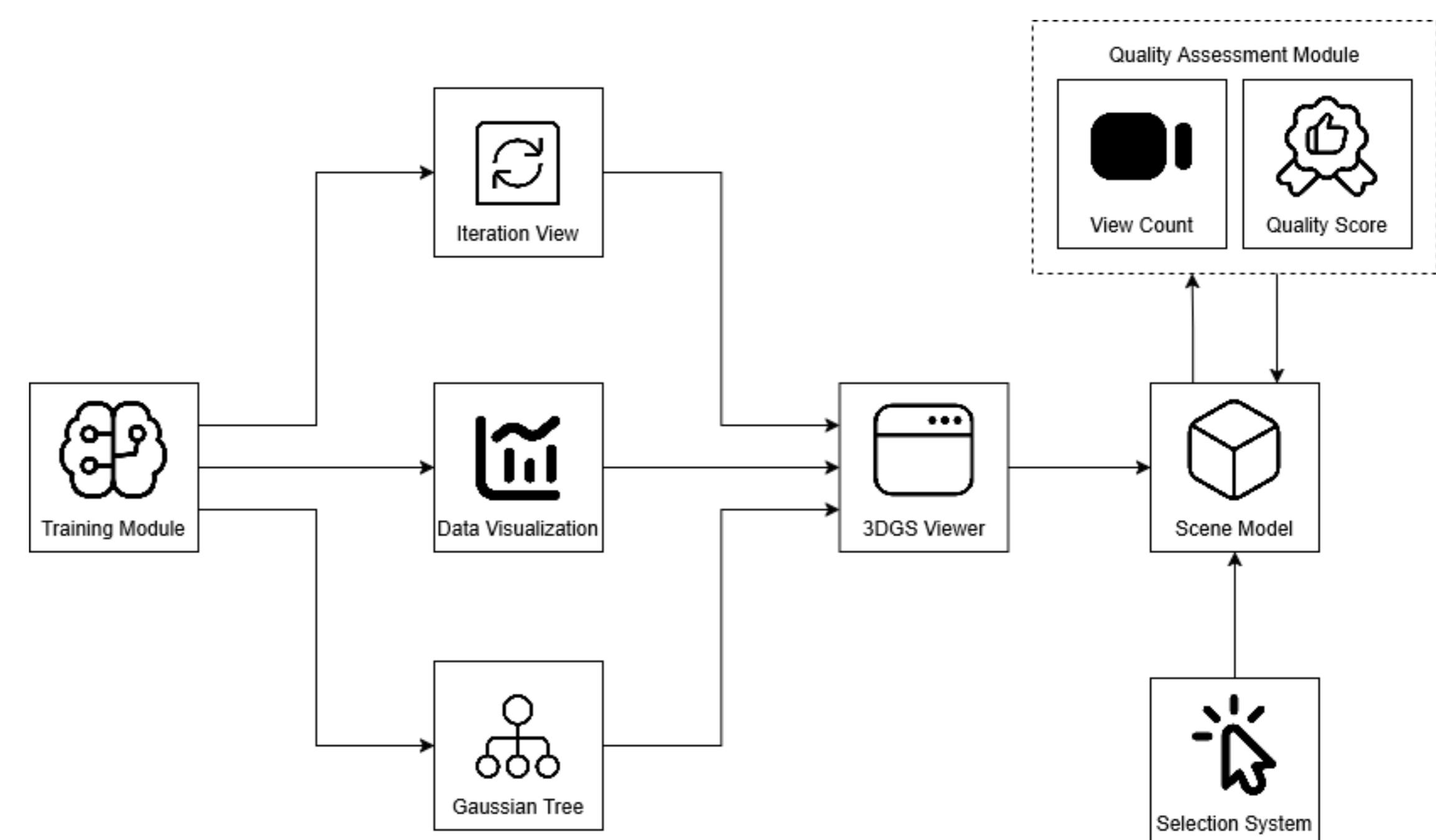


Fig. 2. System Overview of Vis4GS: Iteration View, Training Data Visualization, Gaussian Tree, Quality Score, View Count, and Selection System.

Results and Evaluation

System Performance

Our tool achieves acceptable performance for analysis, on top of the overhead from the original viewer, which is depreciated.

Metric	Dataset	Value
Viewer Load Time	750K Gaussians	135 s
Loading New Iteration Time	60K → 750K Gaussians	10 s
3D Viewer Framerate (500 iter)	60K Gaussians	120 FPS
3D Viewer Framerate (30000 iter)	750K Gaussians	110 FPS
Reduction of FPS per Log Chart	-	30 FPS

Table 1. Viewer can achieve respectable FPS even with larger datasets.

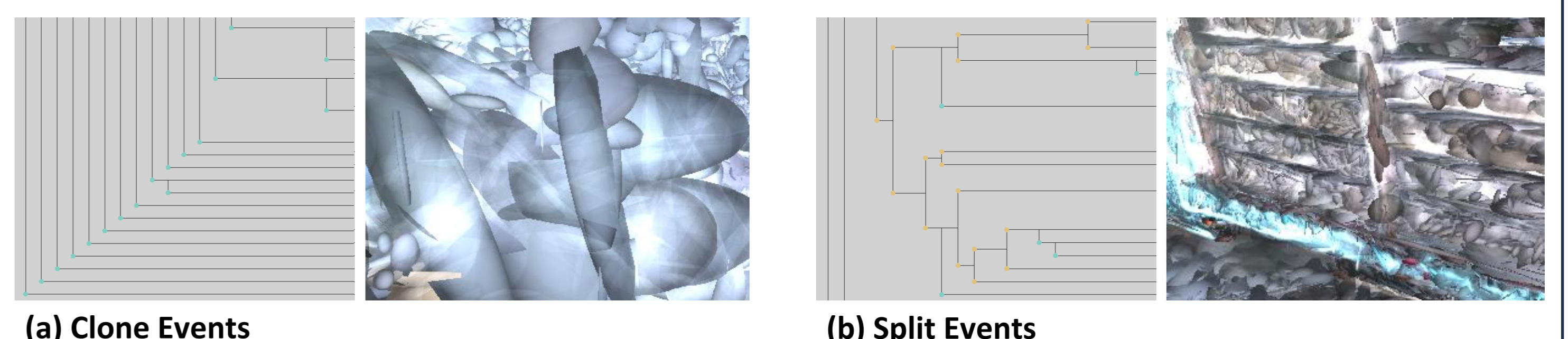
Selection Mode	Time (ms)
Single Selection	~0.70
Brush Selection	~0.63
Frame Selection	~0.91
Ray Casting Selection	~20

Table 2. Selection mode achieves fast performance for fluid interaction

Case Study 1: Analyzing Densification Strategy with Gaussian Tree

The Level-of-Detail (LoD) feature for the Gaussian tree is crucial for managing visual clutter. By using the mapping feature to link tree nodes to the 3D scene, users can gain clear insights.

For example, they can observe how (a) low-detail areas (like floors) are filled with large, cloned Gaussians, while (b) high-detail objects are composed of many small, split Gaussians.



Case Study 2: Guided Refinement Diagnosis in 3DGS Scenes

The highlighted region in the original render appears overly smooth and lacks detail. The Taming3DGS importance score marks this area as high-importance, indicating that a few oversized splats are dominating appearance and the region requires densification. Meanwhile, the View Count map shows low camera coverage, and Selection confirms large splat scales, explaining why this region reconstructs poorly.

