INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

INTERACTIVE SEGMENTATION OF GAUSSIAN MODELS

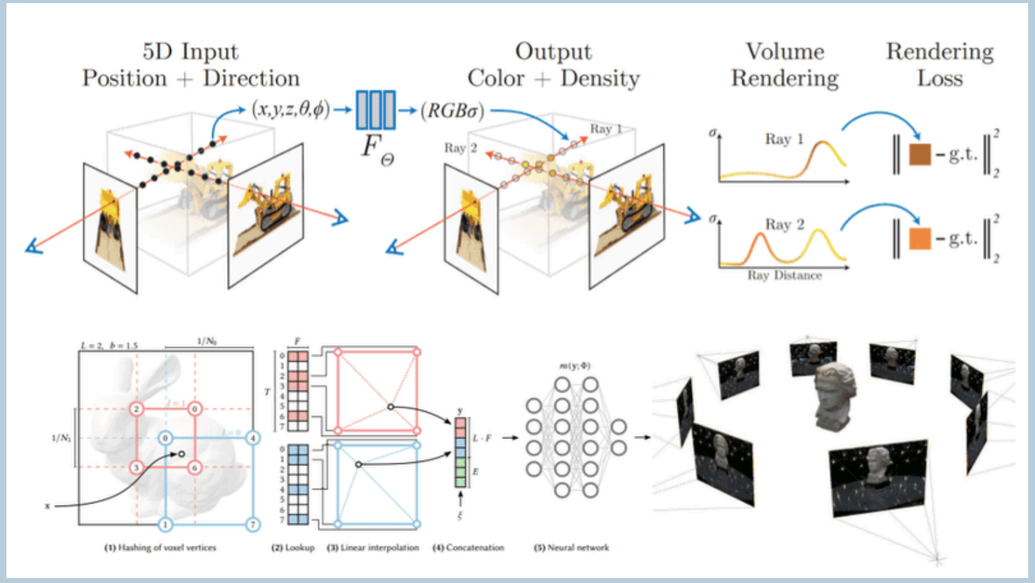
TEAM: 38

P Harshavardhan

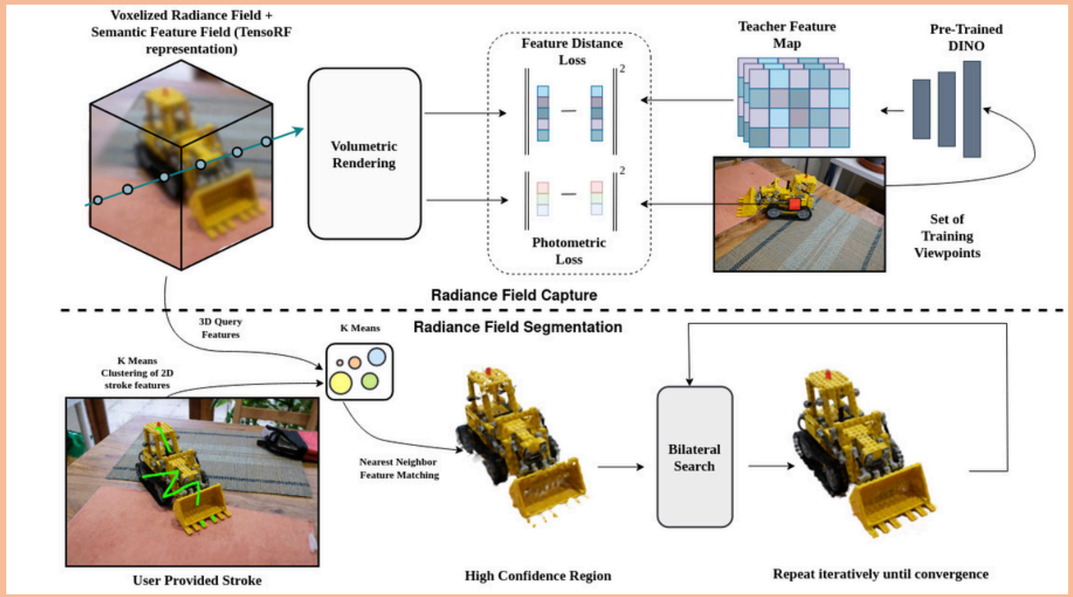
P. V. Seshadri

NERF PAPERS

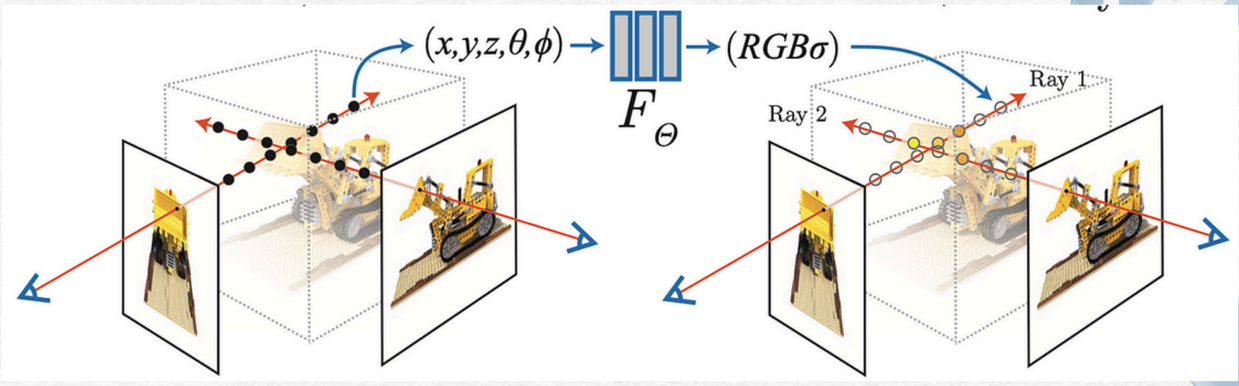
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis



ISRF: Interactive Segmentation of Radiance Fields



VOLUMETRIC RENDERING

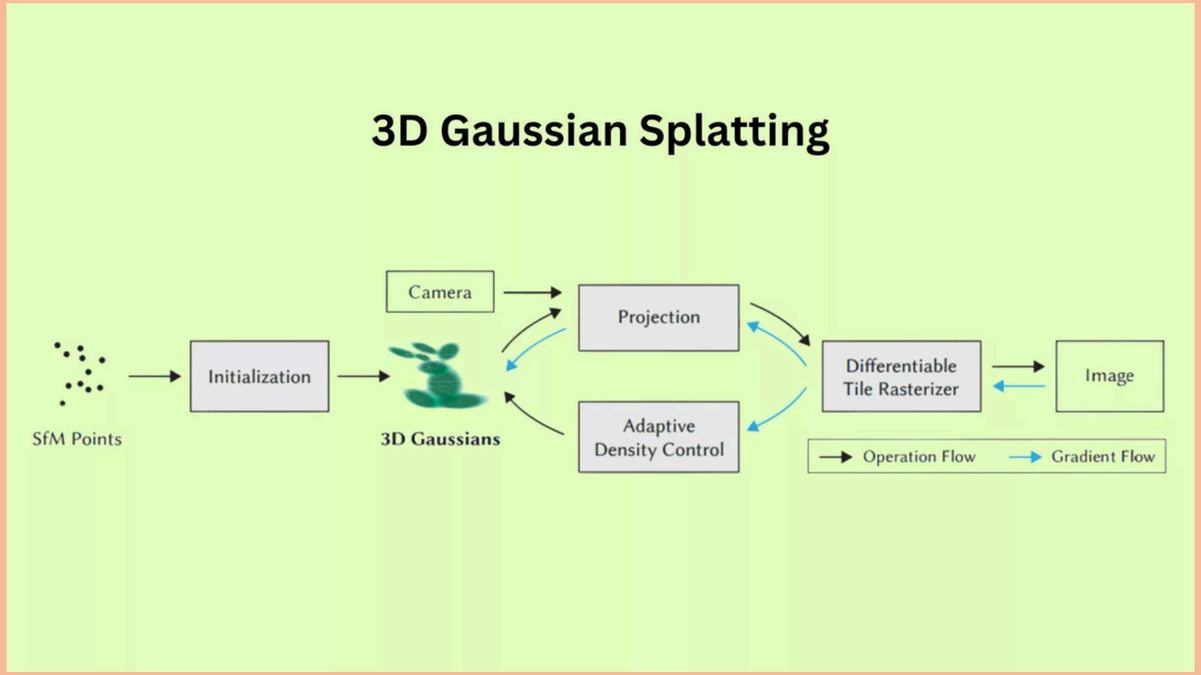


$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right).$$

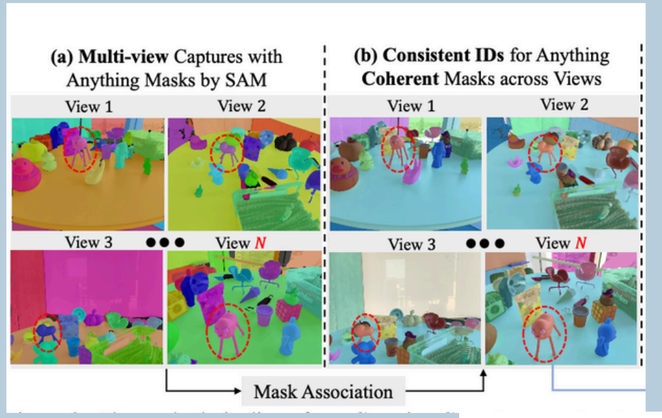
$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right),$$

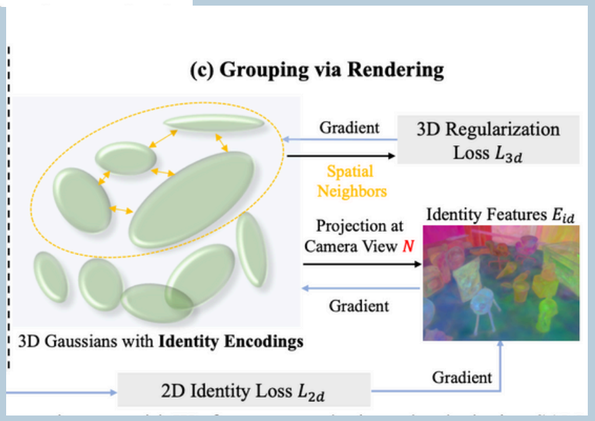
GAUSSIAN PAPERS

3D Gaussian Splatting for Real-Time Radiance Field Rendering



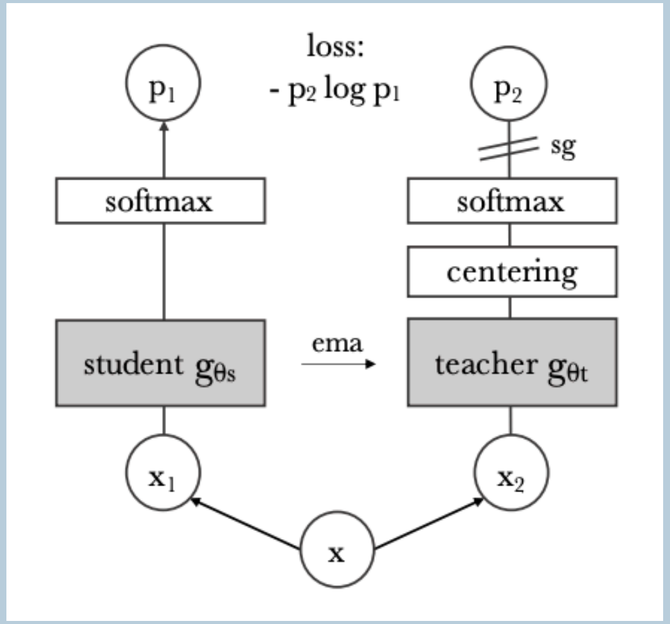
Gaussian Grouping: Segment and Edit Anything in 3D Scenes





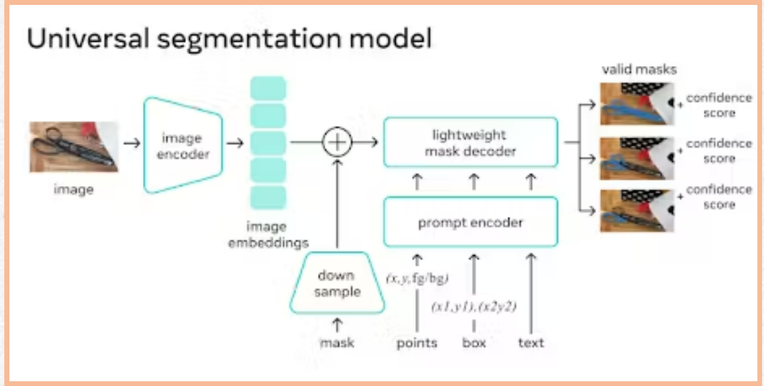
ADDITIONAL REFS

Meta DINO: Emerging Properties in Self Supervised Vision Transformers



Meta SAM: Segment Anything Model

SAM is a promptable segmentation system with zero-shot generalization to unfamiliar objects and images, without the need for additional training.



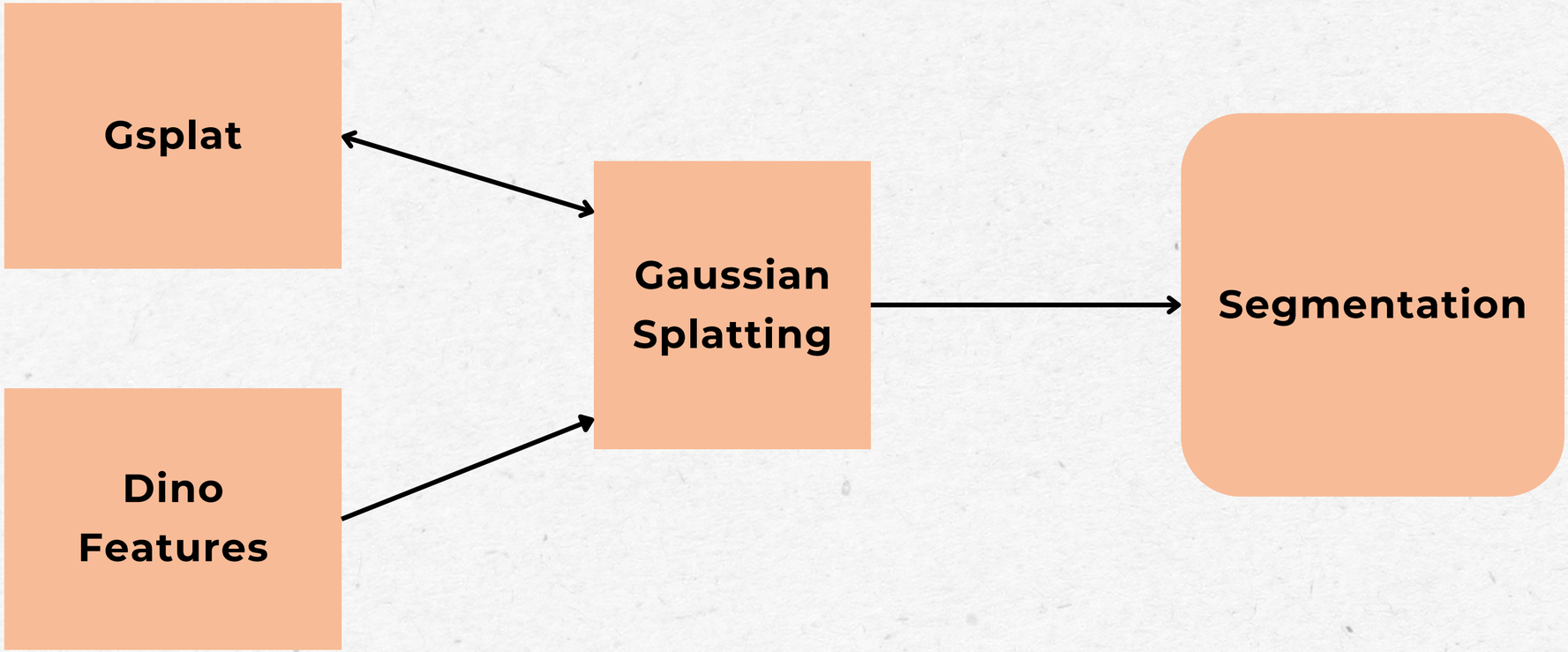
Gsplat: Nerfstudio, UC Berkley

Gsplat is an open-source library for CUDA accelerated rasterization of gaussians with python bindings.

Supports more than 3 channels of training, up to 16 channels by default, in place of 3 channel training in original diff-gaussian-rasterizer limited to 3 channels.

This library provides simple to replace and use python bindings as a replacement for the diff-gaussian-rasterizer.

PROPOSED PIPELINE



EXPERIMENTS AND OUTCOMES

GSPLAT

Replacing diff-gaussian-rasterization with gspalt for training the gaussian model with more than 3 channels directly did not work. No references available who has used gsplat before, and any tries did not fare well

ISRF LOSS

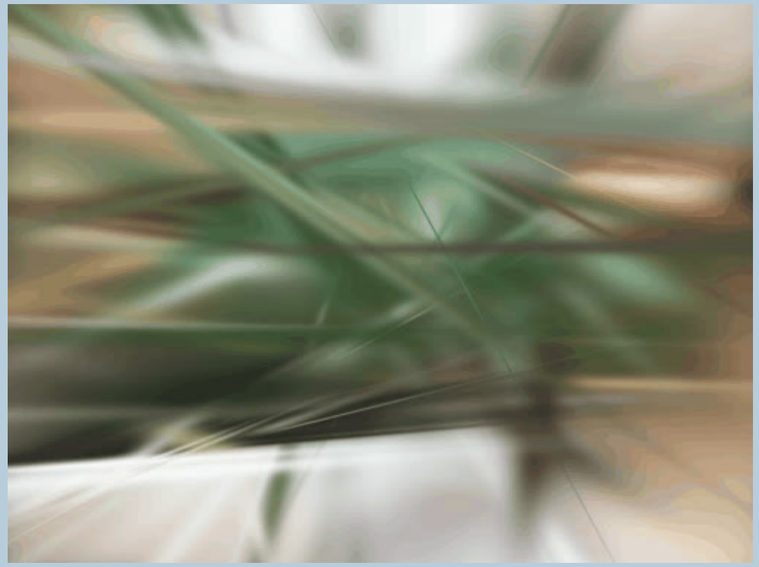
Trainging over 3 channels of data with diff-gaussian-rasterization had a lot of bugs and loss calculation similar to ISRF did not result in convergence of any channels. The reason has not been found out yet, but should likely be a reason of opposing gradients.

INDEPENDENT TRAINING

Training each of the 64 channels in batches of 3 channels at a time showed clear convergence. repeating the training for all the channels and finally combining the learnt features would give final results as expected

EXPERIMENTS AND OUTCOMES

GSPLAT



ISRF LOSS

RGB -> rgb channels
Feats -> Feature channels

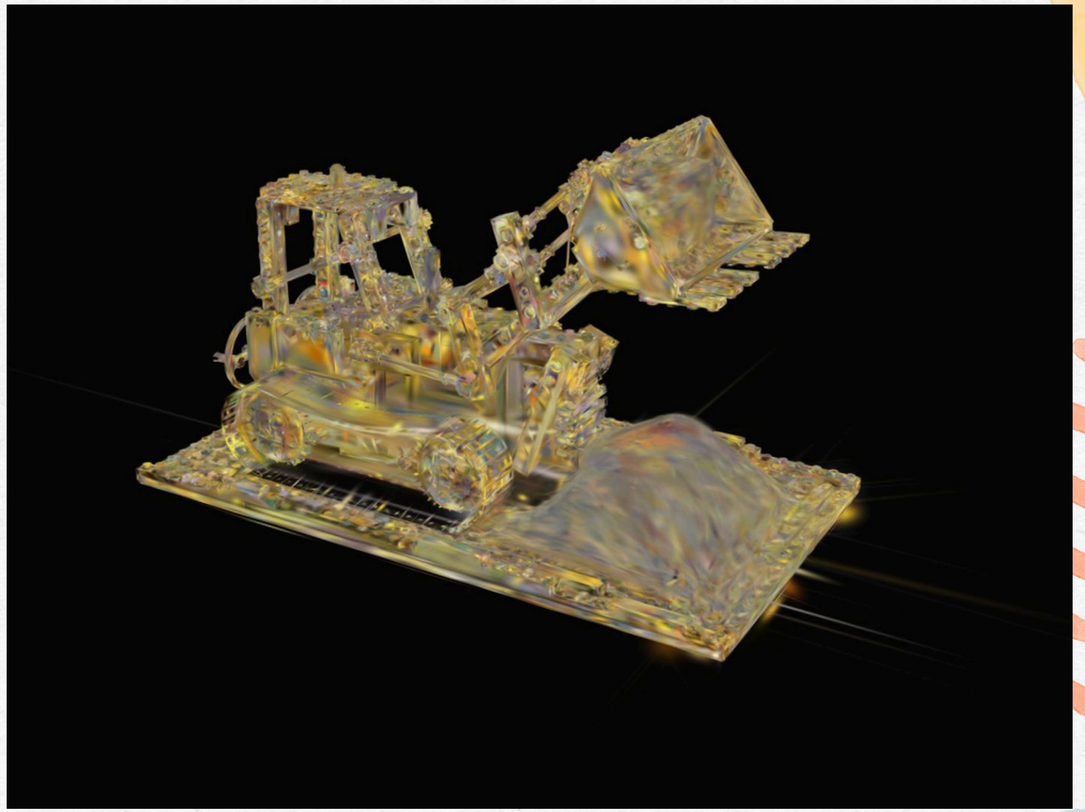
loss1 = loss(RGB)
loss2 = loss(Feats)

Final loss = (1000 * loss1
loss2) / 1001

This weighted nature of loss pushes structure to be learnt from RGB while also learning features.

INDEPENDENT TRAINING

Training each of the 64 channels in batches of 3 channels at a time showed clear convergence. repeating the training for all the channels and finally combining the learnt features would give final results as expected



WHAT WORKED

01

‘diff-gaussian-rasterization’ that came along with gaussian splatting can be used for more than 3 channels, other git users have pushed up to 6 channels. we can train subsets of the dino features channels one at a time for backing in the features rather than all at once.

02

Loss function similar to ISRF gave good results for convergence when trained on gsplat. however due to non feasibility of transfering gaussian splatting to gsplat closed this option. Yet this option can be explored with modification to gaussian training pipeline to properly use the gsplat api.

03

Dino feature based segmentation without distance metrics works fine as long as intention is to segment out all the objects of similar features. However introducing distance based cutoff will make sure the segmentation is object specific (segments out each object separately)

FINAL TRAINING METHOD

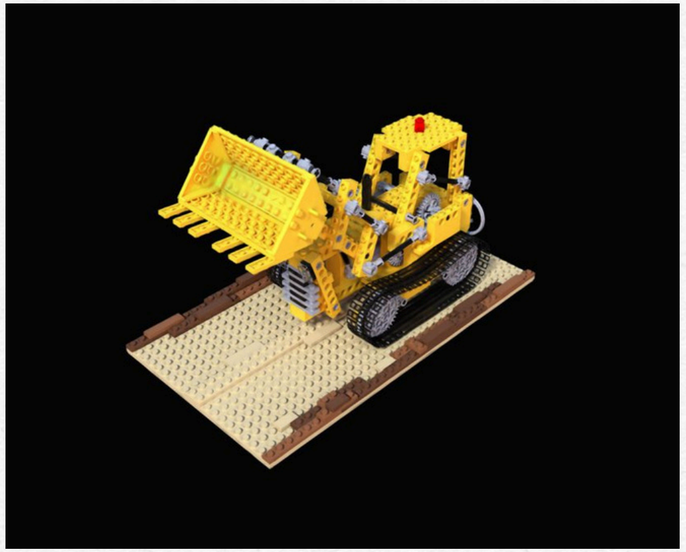
First, train the gaussian model as usual in the RGB color space, with densification.

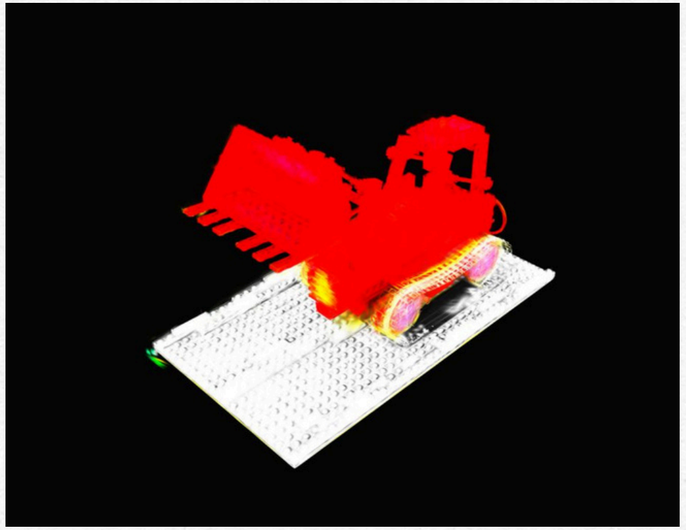
Secondly, freeze all the gradients in the entire model, including xyz, rotations, scales, opacities etc.

Thirdly, create 3 new channels for dino's first three channels and train for learning with frozen values of xyz, rotations, scales etc.

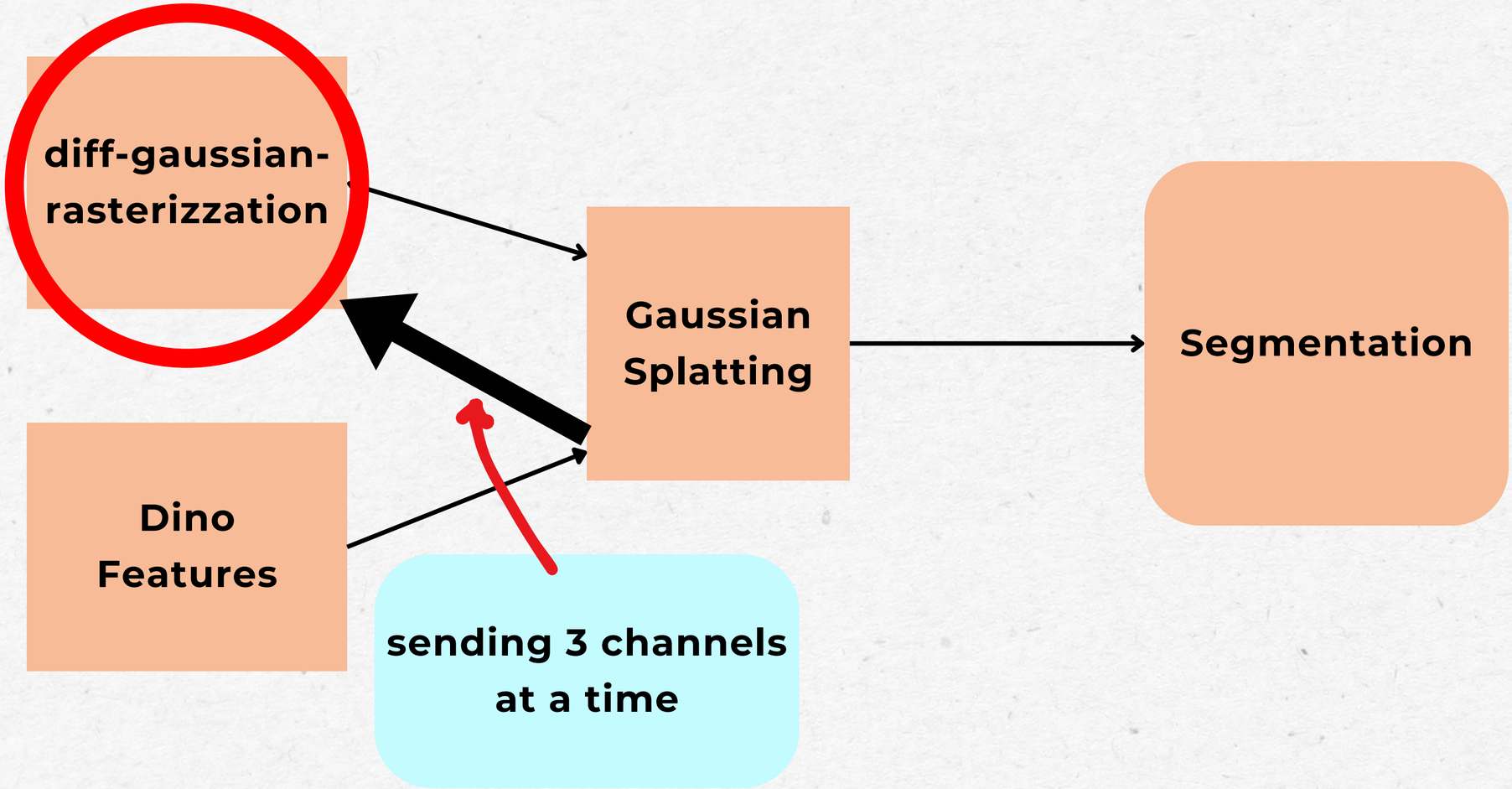
Repeat this training for all the channels of dino required until convergence or required iterations.

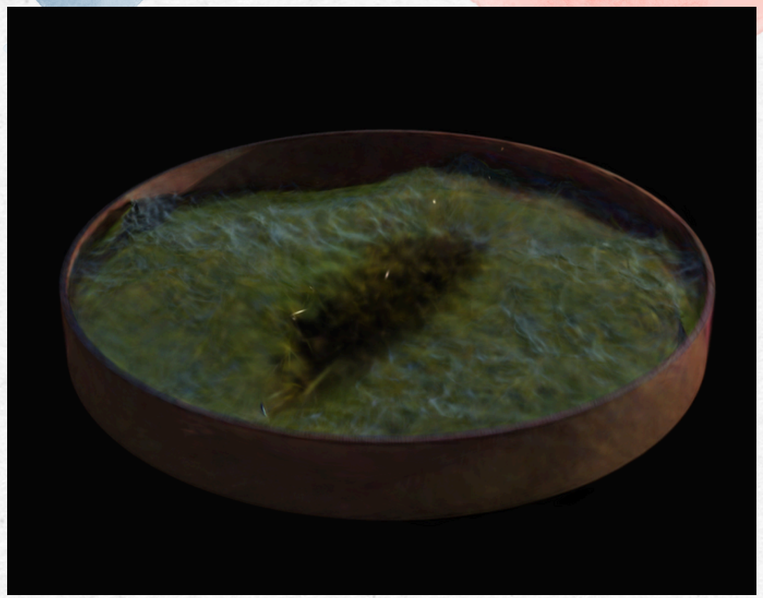
Finally, concatenate all the learnt dino features to get a single tensor of learnt dino values.





CURRENT PIPELINE





WHAT IS LEFT

01

Segmentation isn't designed for interactive use, meaning it's not meant to respond to or change with user actions in real-time. It typically involves analyzing data to divide it into distinct parts, but once set, these divisions don't adapt based on user input. If you want to change the segmentation, you'd need to redo the analysis or create a new one.

02

Region growing-based clustering allows for clustering data by expanding from a seed point, adding neighboring points if they meet specific similarity criteria. This method builds clusters by progressively incorporating adjacent points, resulting in cohesive regions or groups.

03

No single metric exists to accurately gauge segmentation quality, as the ideal measure varies depending on the context and goals of segmentation. Evaluating segmentation often requires a combination of metrics to assess accuracy, precision, and consistency in dividing data into meaningful groups.

UI: WHY IS IT LEFT

- Pygui runs locally not on server. Could not use ADA for the pipeline as ADA does not grant `sudo` permission for X11 forwarding.

- We required a system with atleast 8GB of VRAM, which was not available over the last couple of weeks.

~~INTERACTIVE~~ SEGMENTATION OF GAUSSIAN MODELS

THANK YOU VERY MUCH!