LUDVIG: Learning-Free Uplifting of 2D Visual Features to Gaussian Splatting Scenes

Juliette Marrie[1,2] Romain Menegaux[1] Michael Arbel[1] Diane Larlus[2] Julien Mairal[1]
[1] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK [2] NAVER LABS Europe

- Problem / objective
  - Off-the-shelf module for transferring 2D features into 3D across a wide range of applications
- Contribution / Key idea
  - 2D-to-3D uplifting approach
    - 2D visual features를 3D GS representation으로 transfer
    - Learning-free, Simple yet effective
  - Graph diffusion process
    - 3D scenes에서 feature refinement
    - combines spatial structure with DINOv2 similarity to generate accurate 3D segmentation masks

- **Motivation**

  ❑ **Previous research**

  1. Image understanding: VFM
  2. 3D scene representation: NeRF, Gaussian Splatting
  3. Large 2D pretrained models 로부터 추출한 Image-level semantics을 NeRF나 GS 3D representation에 통합하는 연구

  ❑ **Limitations**

  1. Optimization dependency:
  3D scene의 모든 training views에 대하여 reprojection error를 최소화하면서 반복 학습 필요

  ❑ **Ours**

  1. 2D visual features / semantic masks 를 3D GS scenes로 바로 uplift 하는 간단한 Inverse rendering 방법 제안
     - Simple and learning-free
     - Computationally efficient and highly effective
     - Adaptable to any feature type
  2. Graph diffusion process for 3D segmentation based on uplifted features
     - Feature refinement in 3D scenes
     - Coarse segmentation inputs을 accurate 3D segmentation masks로 변환

- **Overview**

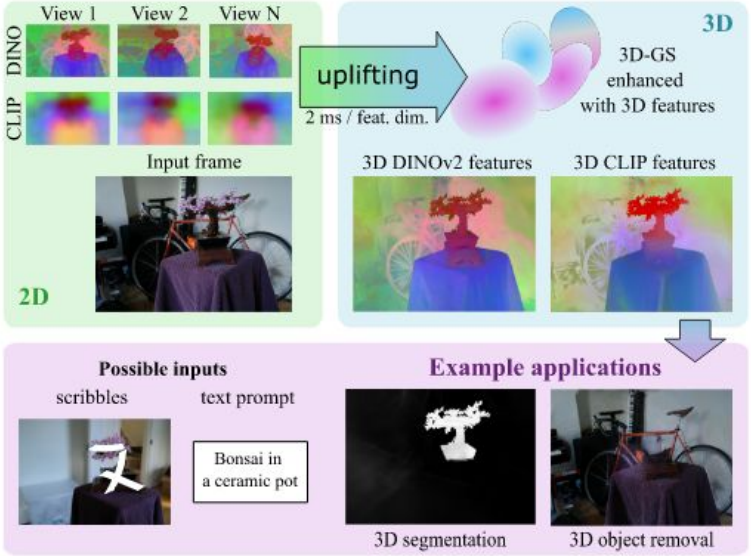


Figure 1. In this paper, we propose a simple, parameter-free method to uplift any 2D visual features (*e.g.*, CLIP or DINO) into a 3D Gaussian Splatting (3D-GS) representation. Uplifting is directly implemented into the rendering process, and takes about 2ms per image and feature dimension. The uplifted features can be leveraged for various 3D tasks, such as segmentation or object removal, based on various inputs, such as scribbles or text prompts.

- **Uplifting 2D visual representations into 3D**
  - **Background on Gaussian Splatting**

➢ **Scene representation**

$$\hat{I}_d(p) = \sum_{i \in \mathcal{S}_{d,p}} c_i(d) w_i(d,p). \qquad (1)$$

2D Rendered image (view d, pixel p)

$$w_i(d,p) = \alpha_i(d,\bar{p}) \prod_{j \in \mathcal{S}_{d,p}, j<i} (1 - \alpha_j(d,p))$$ # alpha-blending

Rendering weights (view d, pixel p) *Gaussian contributions = opacity * density*

➢ **Scene optimization**

- GS optimizes the parameters involved in scene rendering function.

$$\min_{\theta} \frac{1}{m} \sum_{k=1}^{m} \mathcal{L}(I_k, \hat{I}_{d_k,\theta}), \qquad (2)$$ # Reconstruction loss b/w image and rendered image

- **Uplifting 2D visual representations into 3D**
  - **Uplifting 2D feature maps into 3D**

➢ **Uplifting with simple aggregation**

- 각 3D Gaussian의 uplifted feature는 rendering weight에 비례하여 2D features를 가중 평균한 값

$$f_i = \sum_{(d,p)\in\mathcal{S}_i} \bar{w}_i(d,p) F_{d,p}, \quad \bar{w}_i(d,p) = \frac{w_i(d,p)}{\sum_{(d,p)\in\mathcal{S}_i} w_i(d,p)}. \tag{3}$$

3D Uplifted feature

Gaussian i가 view d / pixel p 에 기여한 상대적 영향력

$$\mathcal{S}_i = \{(d,p), i \in \mathcal{S}_{d,p}\}$$

Gaussian i 에 기여하는 view/pixel 집합

- 즉, Uplifting 과정은 rendering linear operator를 transpose 후 normalization 한 값

Rendering to m frames

$$\hat{\mathbf{F}} = W\mathbf{f}, \tag{4}$$

Uplifting from m frames

$$\mathbf{f} = D^{-1}W^{\top}\mathbf{F}. \tag{5}$$

➢ **Gaussian filtering**

- 메모리 효율 위해 하위 50% 중요도의 Gaussians는 pruning함

$$\beta_i = \sum_{d,p\in\mathcal{S}_i} w_i(d,p)$$

Gaussian i가 전체 scene에서 차지하는 기여도의 합

- **Uplifting 2D visual representations into 3D**
  - **Uplifting 2D feature maps into 3D**

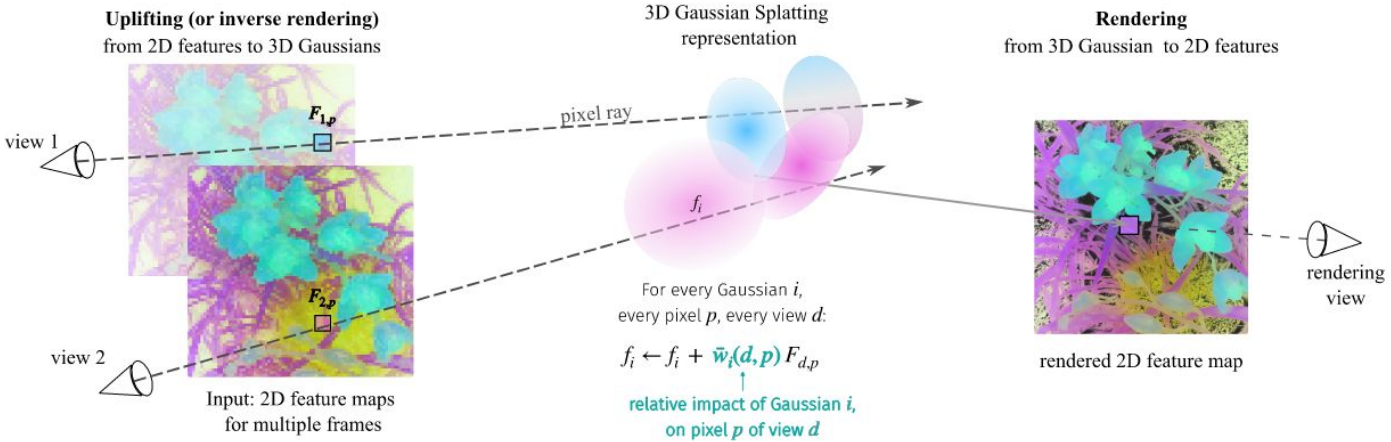


Figure 2. **Illustration of uplifting and rendering.** In the uplifting phase, features $\mathbf{f}$ are created for each 3D Gaussian by aggregating coarse 2D features $\mathbf{F}$ over all viewing directions. For rendering, the 3D features $\mathbf{f}$ are projected on any given viewing direction as in regular Gaussian Splatting. The rendering weight $\bar{w}_i(d,p)$ represents the relative influence of the Gaussian i on pixel p (Eq. (3)).

- **Uplifting 2D visual representations into 3D**
  - **Enriching features by graph diffusion**

➢ DINOv2 내의 semantic 정보를 장면 배치 및 객체 경계와 align하기 위해 3D uplifted features를 diffuse 함

➢ **Graph construction**

- 노드: 3D Gaussians (n개)
- 엣지: matrix A (크기 nxn)
  - 노드 간 geometry 및 DINOv2 features 간 similarity 기반 연결
  - 서로의 노드가 k-nearest neighbors 에 속하면 엣지로 연결

$$A_{ij} = S_f(f_i, f_j)\, P(f_i)^{\frac{1}{2}} P(f_j)^{\frac{1}{2}}, \qquad (7)$$

두 노드 feature 간 local 유사도

- 노드 feature와 관심 객체 feature 간 유사도
- "Gaussian이 관심 객체와 얼마나 관련 있는지"를 나타내는 점수로, diffusion 과정에서 feature가 배경으로 새어나가지 않고 객체 내부에서만 잘 퍼지게 제어하는 역할

➢ **Diffusion on the graph**

- g_0 : 모든 Gaussian의 초기 객체 후보 점수 벡터, diffusion의 출발점 (크기 n)
- Diffusion: 그래프 연결을 따라 신호 퍼짐
- g_T : T diffusion steps 결과, 객체 영역은 값이 크게 남고 배경은 상대적으로 약해져서 노이즈 제거 및 경계 정제

$$g_{t+1} = A\tilde{g}_t, \quad \tilde{g}_t = g_t / \|g_t\|_2, \qquad (8) \qquad (g_t)_{1 \le t \le T}$$

- **From 3D uplifting to downstream tasks**
  - **Multi-view segmentation**

➢ Input: 레퍼런스 프레임 I1에서 segment 원하는 객체의 전경 마스크

➢ Objective: 레퍼런스 마스크 기반으로, 하나 또는 여러 프레임들에서의 2D segmentation 마스크 생성

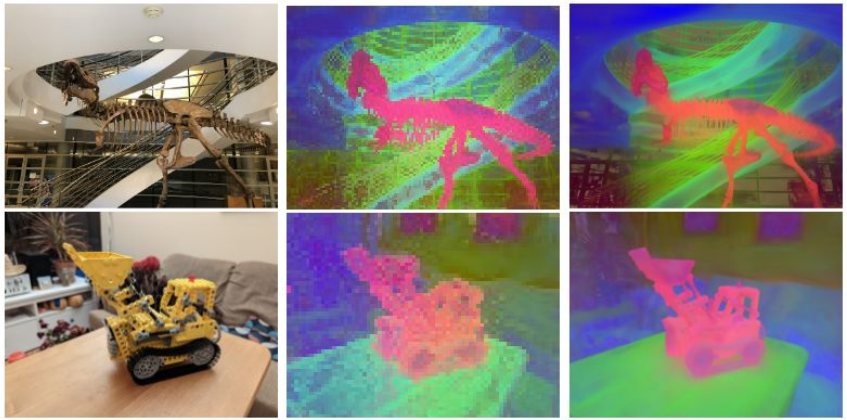
➢ **Segmentation with SAM**

- 1. 레퍼런스 프레임의 SAM 마스크 uplift
- 2. 3D 레퍼런스 마스크를 multiple frames에 reprojection
- 3. Uplift 및 Reproject 반복 후, 결과 예측으로 평균값 사용
- 4. 최종 예측은 3D 마스크를 타겟 프레임에 rendering 하고 thresholding 하여 얻음

➢ **Segmentation with DINOv2**

- 1. Sliding window 통해 DINOv2 사용하여 2D feature maps 생성
- 2. PCA 분해하여 eigenvalues로 features를 re-weight (전경 객체에 집중하는 첫 principal components 강조 목적)
- 3. Uplift 및 Reproject
- 4. 최종 2D segmentation는 projected features와 레퍼런스 프레임의 foreground pixels와의 similarity기반 thresholding 통해 획득

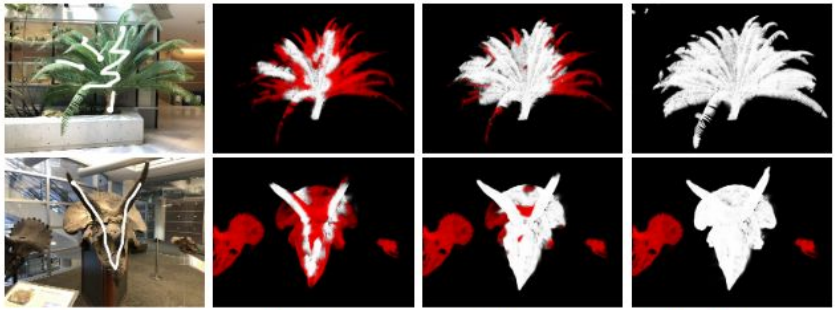
- **Experiments**
  - **DINOv2 feature uplifting**



(a) RGB image (b) Single-view PCA (c) Multi-view PCA

Figure 3. **PCA visualizations.** The DINOv2 patch-level representations (middle) predicted from the RGB images (left) are aggregated into highly detailed 3D representations (right) using Eq. (3).

- **Experiments**
  - **Graph diffusion**



(a) RGB image (b) 3 steps (c) 7 steps (d) 100 steps

Figure 4. **Diffusion process.** 2D projection of the weight vector g_t (white) and unary regularization term (red) at diffusion steps t. The diffusion process filters out unwanted objects that have similar features to the object of interest (such as the two smaller skulls on *horns*, bottom-row), but are disconnected in space. The regularization term (red background) prevents leakage from the object to the rest of the scene (such as through the *fern*'s trunk, top-row).

- **Experiments**
  - **Segmentation results**

Figure B. Segmentation results on NVOS [39] with DINOv2 and SAM.

- **Experiments**
  - **Open-vocabulary object removal**

Figure 5. **Open-vocabulary object removal.** Removing the teddy bear from the CO3D dataset [38], using DINOv2-guided graph diffusion based on 3D CLIP relevancy scores.