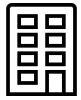
# Towards Realistic Gaussian Avatars

## 2025년도 하계 세미나

***Sogang University***
*Vision & Display Systems Lab, Dept. of Electronic Engineering*

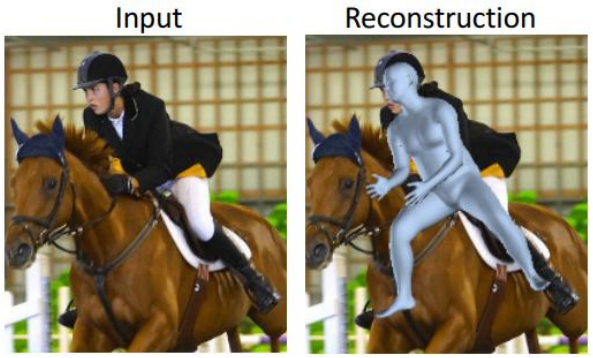
***Presented By***
남준식

# Outline

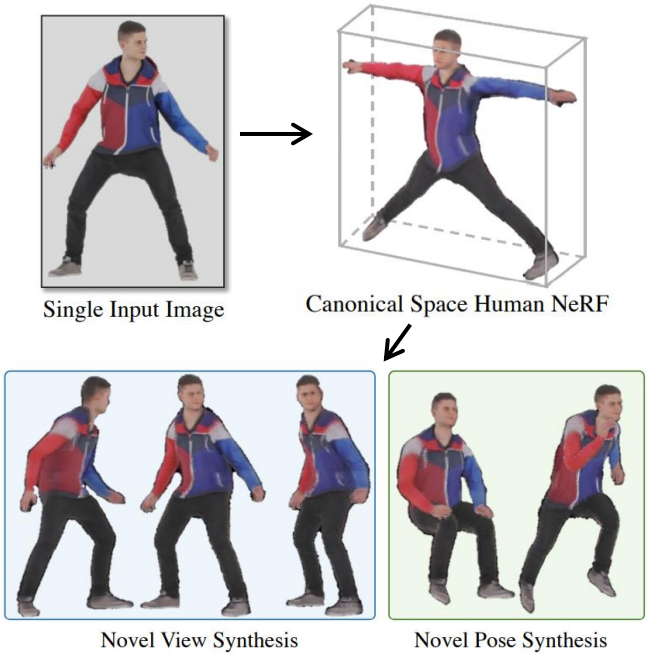
- Background
  - Human Avatar
- 논문 선정의 이유
- Animatable Gaussians: Learning Pose-dependent Gaussian Maps for High-fidelity Human Avatar Modeling
  - CVPR 2024
- HRAvatar: High-Quality and Relightable Gaussian Head Avatar
  - CVPR 2025

# Background

- **Human Avatar**
  - Mesh, Implicit Function, NeRF, 3DGS 와 같은 Representation을 사용하여 3D Space에서 사람을 표현
  - SMPL 기반의 Linear Blend Skinning(LBS)를 통해서 Pose Deformation




Mesh




NeRF


3DGS

# Background

- 논문 선정의 이유
  - **Human Avatar Limitation**
    - Baked Shading : 학습 시 특정 뷰·조명 조건에서 색이 고정되며, 재구성된 3D 표현에 광원 정보가 내재(embedded) 되어버리는 현상.
    - High-frequency loss : 3DGS 표현 특성상, 고주파 성분(세밀한 질감, 날카로운 경계) 포착이 어려워 디테일이 흐려지고 과도하게 매끄러워지는 경향이 있음.

Baked Shading

High-Frequency Loss

# Background

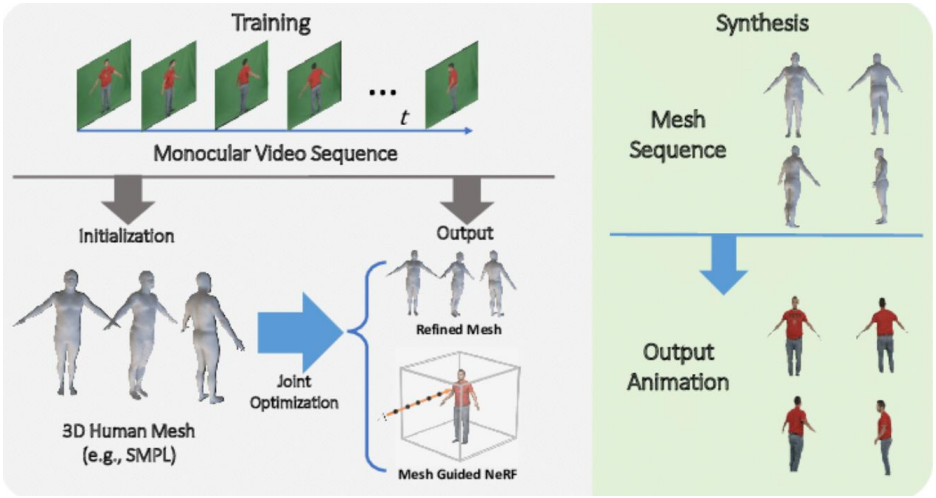
- 논문 선정의 이유
  - **Human Avatar Limitation**
    - Representation constraint : SMPL-X 기반 body-parameterization constraint로 인해서 hair와 clothes 같은 non-body dynamics를 제대로 표현하지 못함

"Animatable Gaussians: Learning Pose-dependent Gaussian Maps for High-fidelity Human Avatar Modeling ."

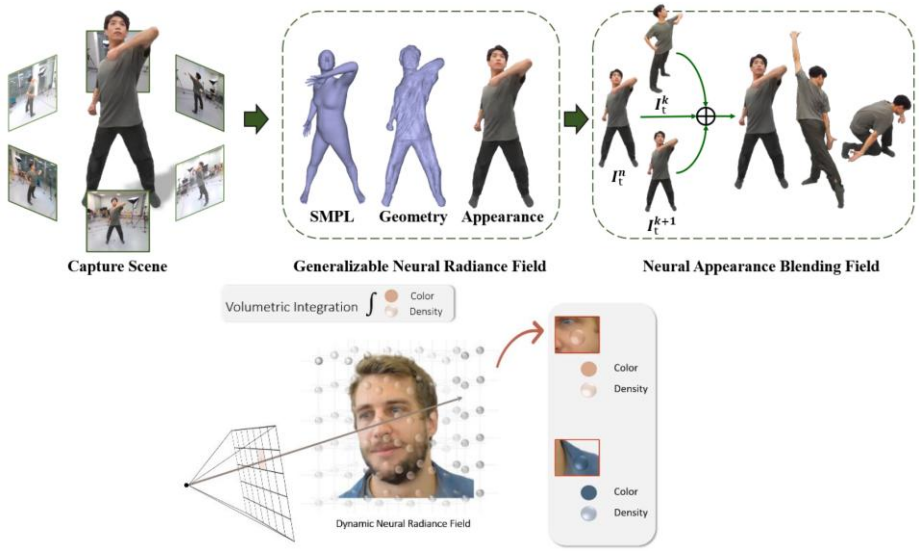
[CVPR, 2024]

# Introduction

- Human Avatar Representation method
  - Explicit Representation(Mesh, Point cloud)
    - 장점 : Explicit Geometry , 빠른 렌더링이 가능
    - 단점 : Dense Mesh 필요, Sparse – view video 기반 모델링에 부적합, 신체 이외의 파트 모델링 불가능
  - Implicit Representation
    - 장점 : 복잡한 기하·재질 표현
    - 단점 : MLP에 의존하기에, High-frequency 표현이 미흡



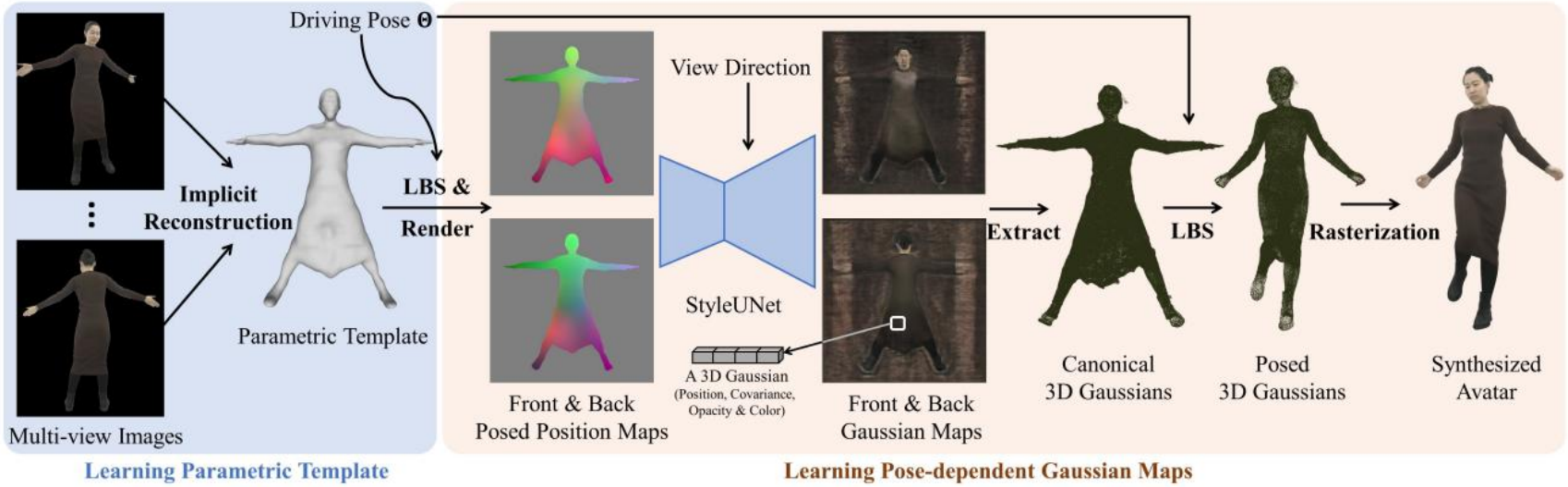

Explicit Representation




Implicit Representation

# Introduction

- Contribution
  - Explicit 3D Gaussian splatting + 2D CNN을 통한 High-fidelity Animatable avatar 제안.
  - Template-guide parameterization을 통해 Personalized template 학습.
  - Driving Signal에 PCA 적용함으로써, Novel Pose에 대한 Robustness 확보.




Framework Overview

# Method

- Learning Parametric Template
  - A-pose를 Canonical Space로 삼아 변형가능한 Template를 만드는 단계
    - Canonical 공간의 Parametric template를 MLP를 통해서 SDF(geometry) 추정
      - 이때 Parametric Template의 pose는 A-pose로 함.
    - SMPL 표면에서 각 vertex의 skinning weight를 normal 방향으로 3D volume 전체에 확산
    - Canonical Space 상의 점 $x_c$ 에 대해서 Root Finding을 통해 학습

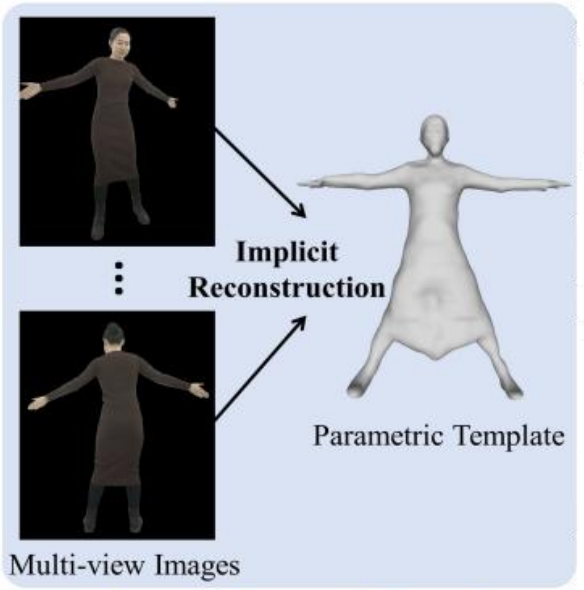
$$\min_{\mathbf{x}_c} \|\mathrm{LBS}(\mathbf{x}_c; \boldsymbol{\Theta}, \mathcal{W}) - \mathbf{x}_p\|_2^2$$

$LBS(\cdot)$ : Canonical → Posed 변환

$W$ : 해당 point 에서의 Skinning weight

$\Theta$ : 현재 Frame의 SMPL pose parameter

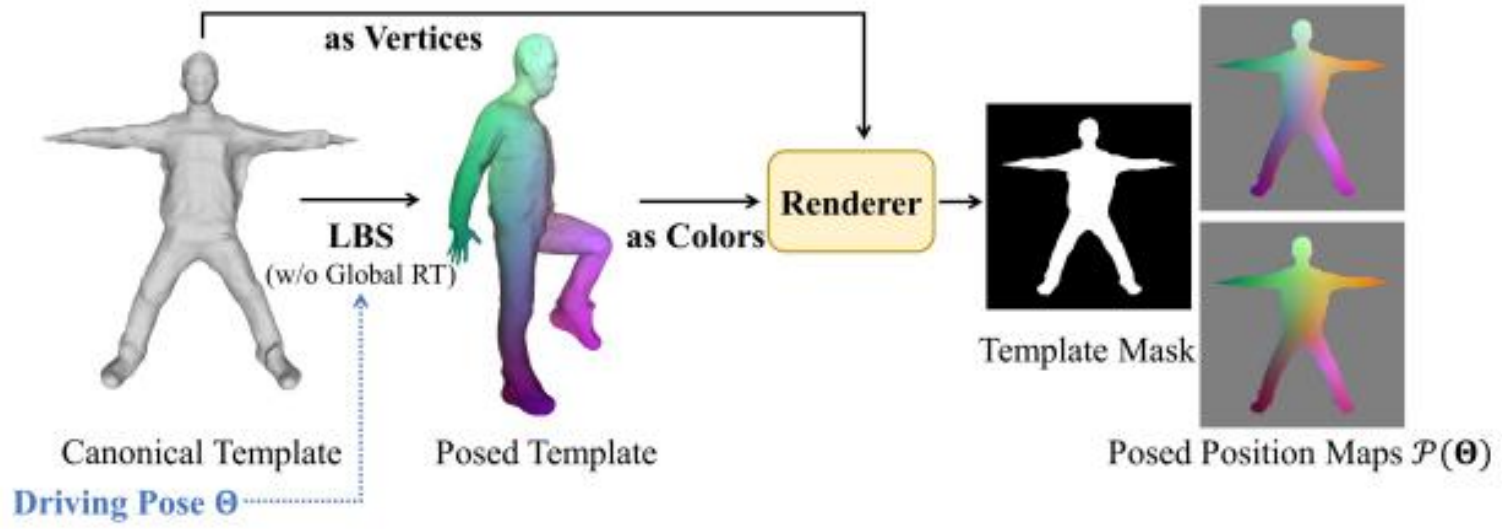
- Marching Cubes를 통해서 Geometry를 추출
- $W$ 통해서 각 vertex에서의 skinning weight를 샘플링




Learning Parametric Template

# Method

- Template-guided Parameterization
  - MLP를 통해서 Color를 학습하는 경우, low-frequency bias 때문에, 옷 주름/손 가락 움직임과 같은 High-frequency detail을 잘 표현하지 못함
    - MLP → 2D CNN을 통해서 Color를 복원
  - 2D CNN을 사용하기 위해서 3D → 2D Parameterization이 필요
    - Canonical Template에 고정된 3D Gaussian들을 Front/Back 2장의 Position map으로 표현



# Method

- Pose-dependent Gaussian Maps
  - Front/Back Position map 을 StyleUNet에 통과함으로써, Pose-dependent Gaussian map을생성
    - StyleUNet을 통해 offset map $\Delta O(\Theta)$ (Position, Covariance, Opacity, Color)를 얻음
      - Offset map을 Position Map에 더함으로써 Gaussian Maps 획득
    - Template Mask를 통해서 3D gaussian 성분들을 Gaussian map에서 추출

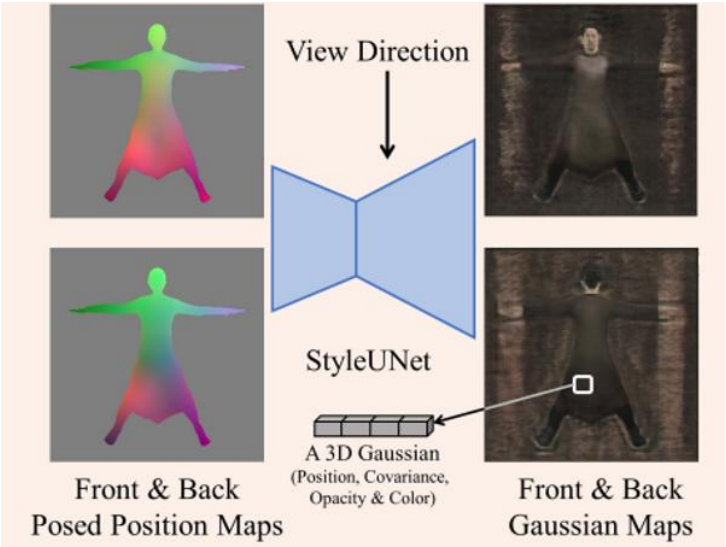


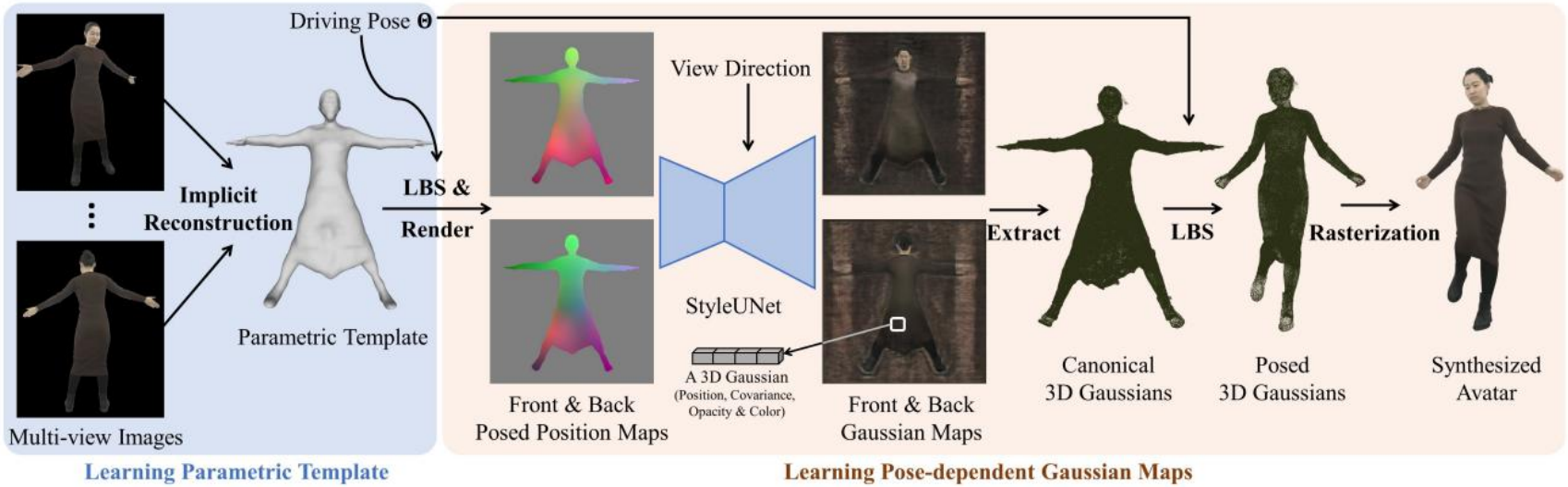

Figure 4. **Canonical 3D Gaussians on side regions and hands.**

# Method

- Loss Function
  - L1 Loss
  - Perceptual Loss (VGG based) - LPIPS
  - L2 Regularization

$$\mathcal{L} = \mathcal{L}_1 + \lambda_{\text{perceptual}}\mathcal{L}_{\text{perceptual}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$$

$$\mathcal{L}_{\text{reg}} = \|\Delta\mathcal{O}(\boldsymbol{\Theta})\|_2^2$$



# Method

- Pose Projection Strategy
  - Novel Pose에 대한 Generalization Strategy
    - 학습 중 얻은 모든 Posed position map을 이어 붙여 행렬 X 에 대해서 PCA를 수행
    - Novel Pose의 Position map을 벡터 X로 변환 후, PCA 공간으로 Projection.

$$\beta = S^T \cdot (X - \bar{X}) \ , \ \beta_i \in [-2\sigma_i, 2\sigma_i] \ \text{clipping}$$

    - Low-dimensional coefficient $\beta$ 에 대해서 Principal components S 를 이용하여 $X_{recon}$ 획득 $X_{recon}$ 을 reshape 하여 novel pose의 position map을 획득

$$X_{recon} = S \cdot \beta + \bar{X}$$

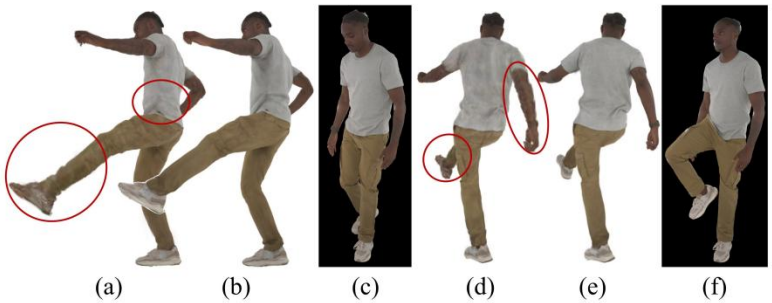



Figure 10. **Ablation study of the pose projection strategy.** (a,d) and (b,e) are the animation results without and with the pose projection strategy, respectively. (c,f) are the reference images with the closest pose in the training dataset.

# Method

- Experiment
  - Thuman-4.0 ( 3 sequence, 24 view )
  - ActorsHQ ( 5 sequence, 47 view)
  - 각 시퀀스를 시간축으로 나누어서 Training Chunk, Test Chunk로 구분
    - Training chunk의 경우 1500 ~ 3000 frame으로 구성
- Baseline ( NeRF-based Avatar )
  - PoseVocab, SLRF, ARAH, TAVA: Body–only
  - AvatarRex : Full-body
- Metric
  - PSNR, SSIM, LPIPS, FID

Table 1. **Quantitative comparison with state-of-the-art body-only avatars.**

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Ours | **28.0714** | **0.9739** | **0.0515** | **29.4831** |
| PoseVocab [45] | 26.3784 | 0.9707 | 0.0592 | 49.4541 |
| SLRF [106] | 26.9015 | 0.9724 | 0.0600 | 52.0613 |
| ARAH [86] | 22.3004 | 0.9616 | 0.1075 | 90.6077 |
| TAVA [43] | 26.8019 | 0.9705 | 0.0915 | 96.3474 |

Table 2. **Quantitative comparison with AvatarReX.**

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Ours | **30.6143** | **0.9803** | **0.0290** | **13.2417** |
| AvatarReX [107] | 23.2475 | 0.9567 | 0.0646 | 31.1387 |

Table 3. **Comparison on animation speed.** These framerates are evaluated on a PC with one RTX 3090 when rendering images at a resolution of 1024 × 1024. We highlight the highest and second-highest framerates.

| Method | TAVA [43] | ARAH [86] | SLRF [106] | PoseVocab [45] | AvatarReX [107] (PyTorch) | AvatarReX (TensorRT) | Ours (PyTorch) |
|---|---|---|---|---|---|---|---|
| Framerate (FPS) ↑ | 0.003 | 0.07 | 0.16 | 0.20 | 0.03 | 25 | 10 |

# Method

- Qualitative Result
  - NeRF 기반의 Avatar가 잘 포착하지 못하는 High-frequency details 잘 포착하는 것을 확인
  - Novel Pose에서도 Avatar의 Robustness를 확인

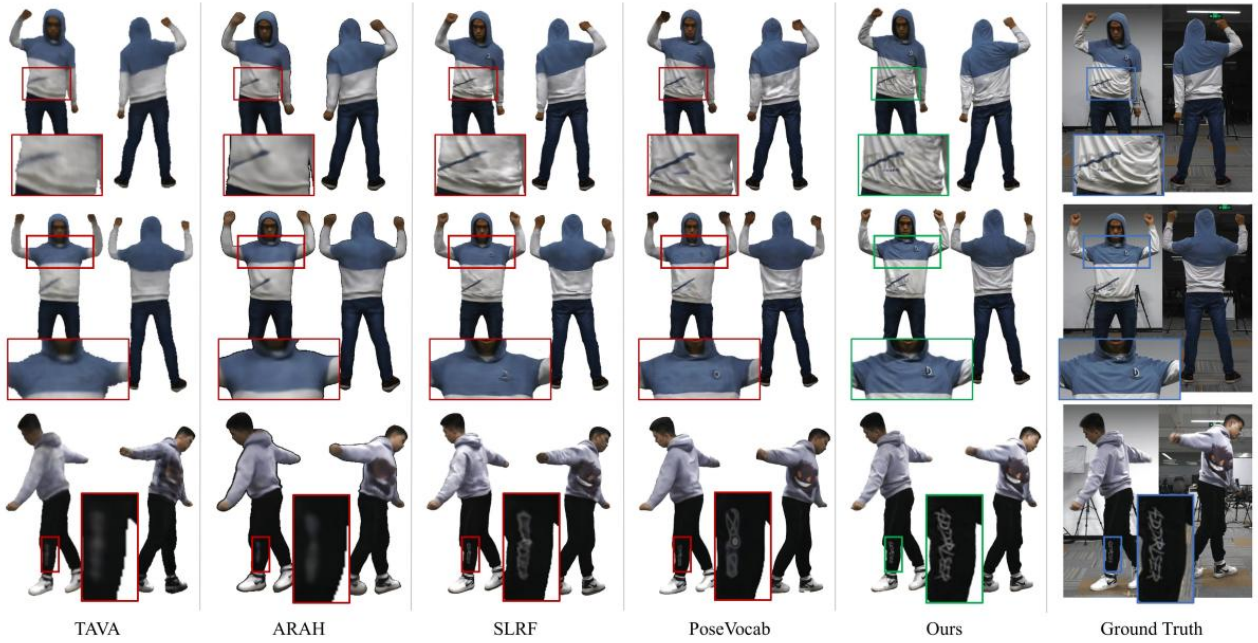




Figure 6. **Qualitative comparison with state-of-the-art body-only avatars on novel pose synthesis.**

# Method

- Ablation study
  - Parametric Template

| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Parametric Template | **31.2183** | **0.9858** | **0.0344** | **36.9905** |
| SMPL-X | 30.5241 | 0.9842 | 0.0401 | 47.5066 |

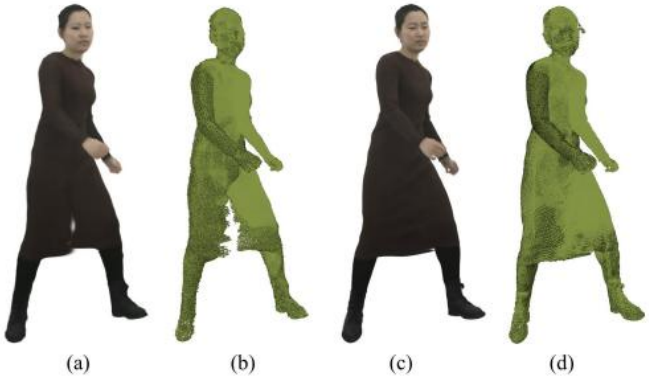



Figure 8. **Ablation study of the parametric template.** (a,b) Rendered results and 3D Gaussians using SMPL-X. (c,d) Rendered results and 3D Gaussians using the character-specific template.

  - Backbones

| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| StyleUNet [83] | **29.3127** | **0.9664** | **0.0378** | **27.3143** |
| U-Net [70] | 26.4255 | 0.9435 | 0.0507 | 31.3838 |
| MLP | 26.8961 | 0.9497 | 0.0650 | 87.0793 |


Figure 9. **Comparison between representations with different backbones on training pose reconstruction.**

# Method

- Ablation study
  - Pose Projection Strategy

| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| w Pose Proj. | **24.9932** | **0.9285** | **0.0685** | **45.6266** |
| w/o Pose Proj. | 23.5594 | 0.9189 | 0.0792 | 59.9083 |

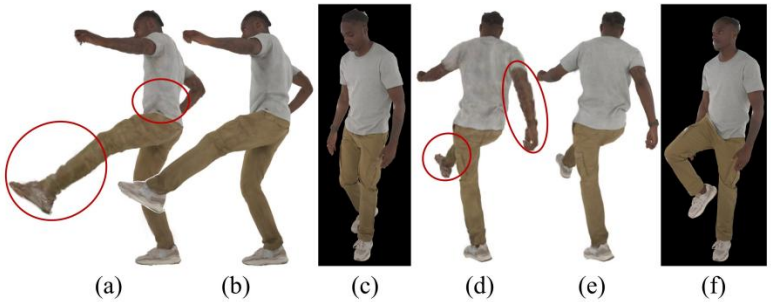



Figure 10. **Ablation study of the pose projection strategy.** (a,d) and (b,e) are the animation results without and with the pose projection strategy, respectively. (c,f) are the reference images with the closest pose in the training dataset.

  - Num of views

| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| 3 Views | 30.6123 | 0.9807 | 0.0306 | 11.3066 |
| 6 Views | 30.3565 | 0.9803 | 0.0310 | 10.9966 |
| 14 Views | 30.7622 | 0.9816 | 0.0297 | 10.6744 |

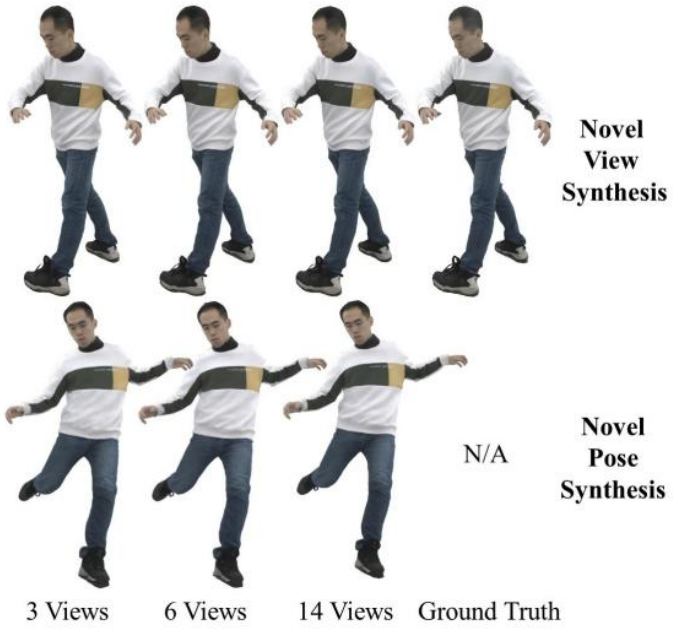



Figure C. **Animation results trained with different numbers of views.**

# Method

- Limitation
  - Hair에 대한 Modeling 실패
  - Clothes, Hairs, Hands를 분리하지 않은 채 Gaussian Representation을 진행
    - Clothes change task 불가능

- Conclusion
  - 3DGS + 2D CNN을 통한 Avatar 를 통해서 Detail한 Human representation 및 Novel pose에 대해서 Realistic 한 Garment Dynamic를 구현

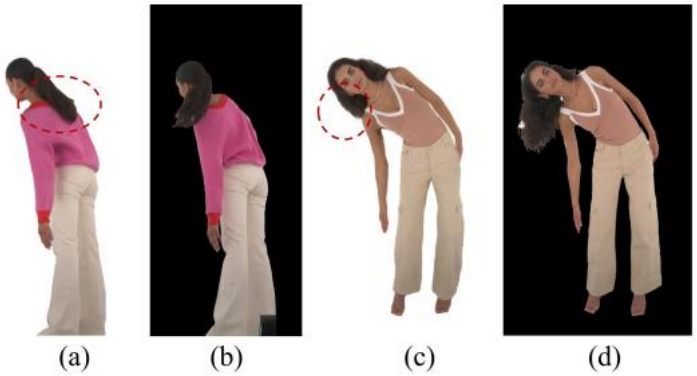


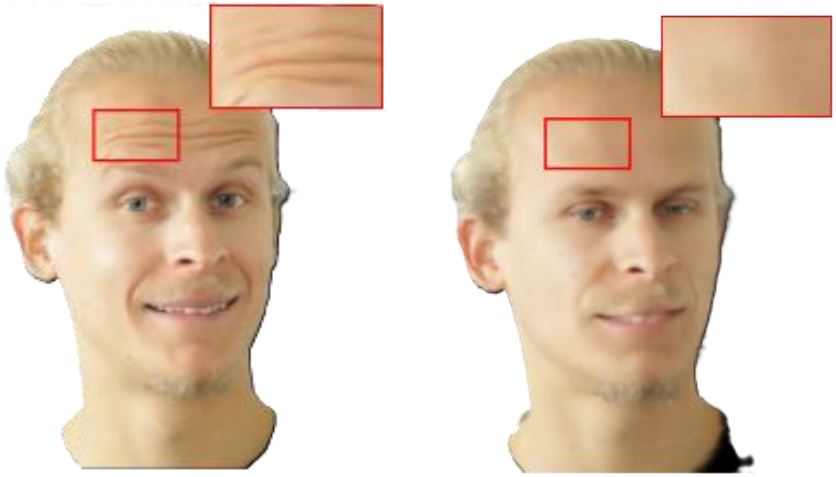

Figure D. **Failure cases.** (a,c) Animation results by our method, (b,d) ground-truth images. Our method fails to model the motion of hairs.

"HRAvatar: High-Quality and Relightable Gaussian Head Avatar "

[CVPR, 2025]

# Introduction

- 3DGS-based Animatable Head Avatar
  - Real-time rendering & Animate 가능
  - Limitation
    - Limited deformation flexibility
    - Inaccurate expression tracking
    - Unable to produce realistic relighting effects

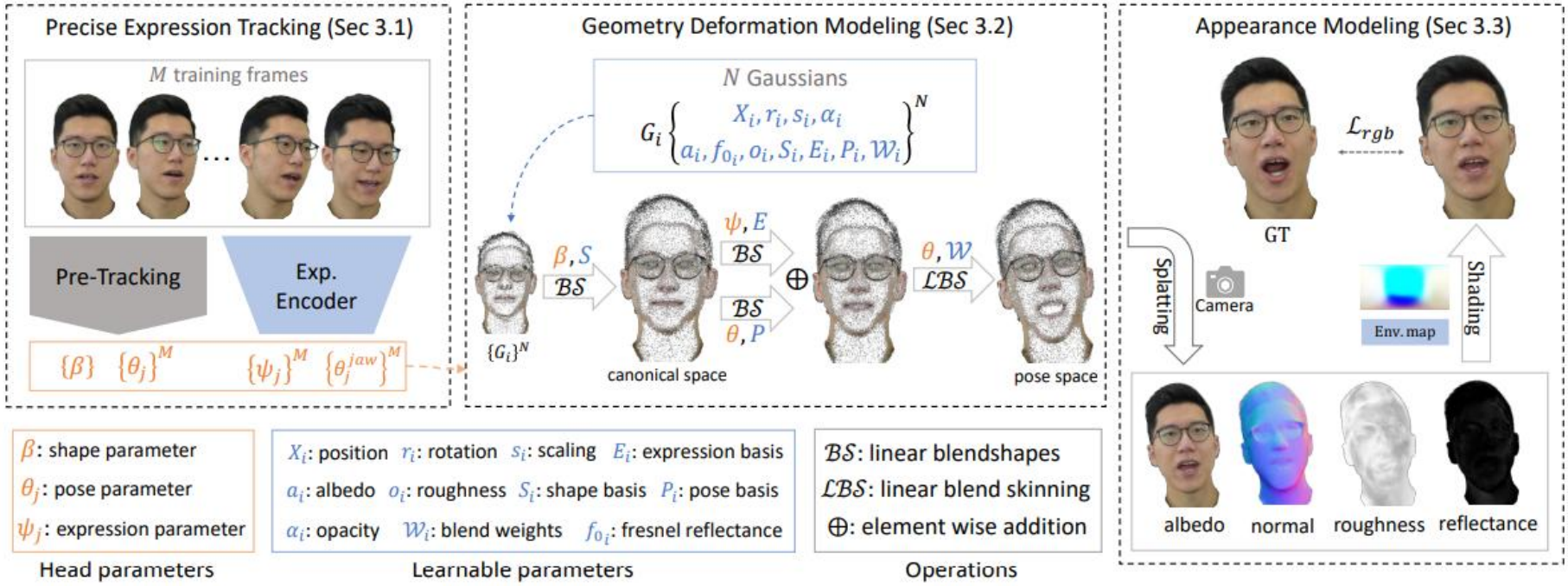


Limited deformation flexibility

Unable to produce realistic relighting effects

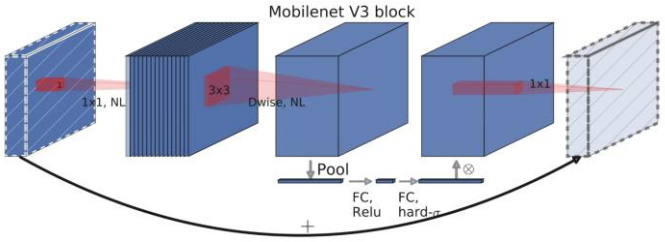
# Introduction

- Contribution
  - Learnable Blendshape & LBS strategy
  - Face Expression Encoder for reducing tracking error
  - Lighting with Physical-based shading model



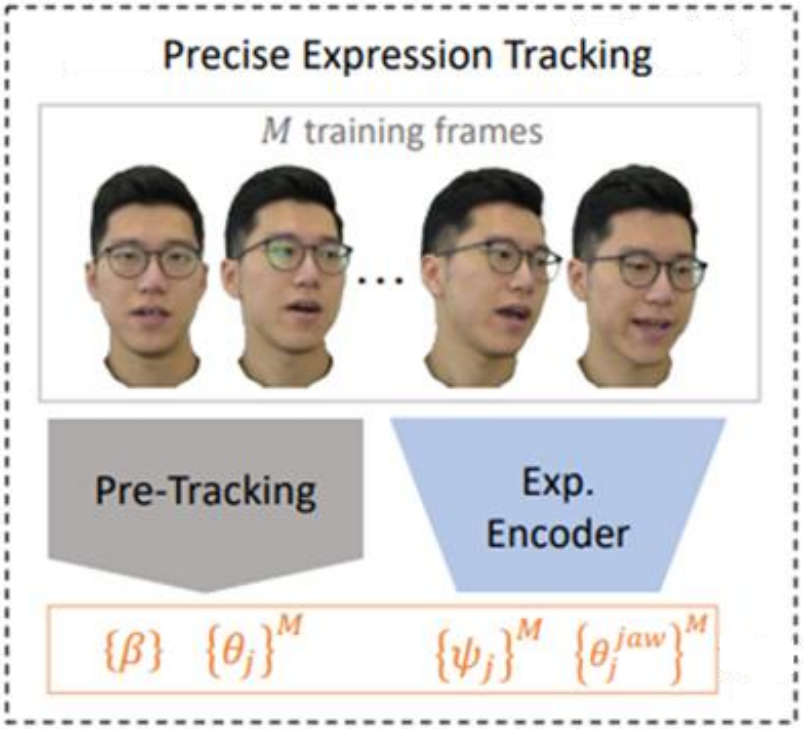
# Method

- Precise Expression Tracking
  - Accurate Expression Tracking을 위한 Encoder $\mathcal{E}$ 제안
    - $\mathcal{E}(I) = \psi, \theta^{jaw}$
    - $\mathcal{E}(\cdot)$는 MobileNetv3 + MLP로 이루어진 구조
      - $\mathcal{E}(\cdot)$은 End-to-End로 학습




    - Pre-Tracking model → DECA
      - $\beta$ (shape) 의 경우, 사람 고유의 특성이므로 모든 Frame에서 공유

$\beta$: shape parameter
$\theta_j$: pose parameter
$\psi_j$: expression parameter
Head parameters



# Method

- Geometry Deformation Modeling
  - LBS의 topology constraint → Learnable Basis & skinning Weight 도입
    - $X_p = \mathcal{LBS}(X_e, \mathcal{J}(\beta), \mathcal{W}) = R_{lbs}X_e + T_{lbs}$ → Skeleton 기반의 변형
    - $X_e = X_c + \mathcal{BS}(\beta, S) + \mathcal{BS}(\psi, E) + \mathcal{BS}(\theta, P)$ → Parameter 기반의 변형

$$\mathcal{BS}(\beta, S) = \sum_{m=1}^{|\beta|} \beta^m S^m$$ → 사람 개개인의 체형 차이

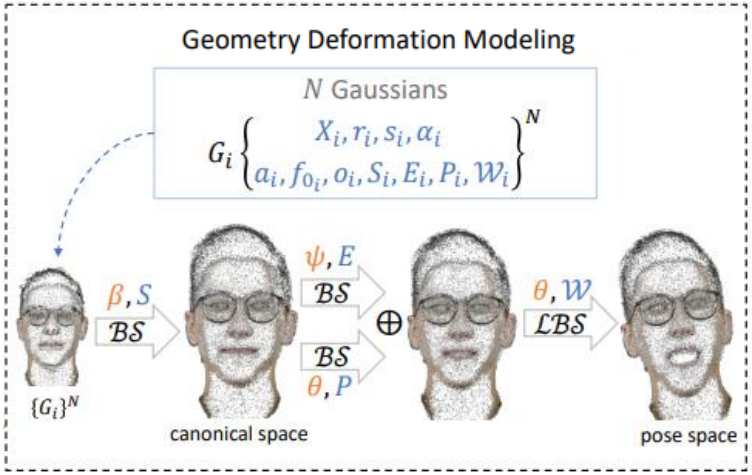
$$S = \{S^1, \dots, S^{|\beta|}\} \in R^{N\times 3\times|\beta|}$$

$$\mathcal{BS}(\psi, E) = \sum_{m=1}^{|\psi|} \psi^m E^m$$ → 얼굴의 표정 변화

$$E = \{E^1, \dots, E^{|\psi|}\} \in R^{N\times 3\times|\psi|}$$

$$\mathcal{BS}(\theta, P) = \sum_{m=1}^{|\theta|} \theta^m P^m$$ → 자세에 따른 비선형 보정

$$P = \{P^1, \dots, P^{9K}\} \in R^{N\times 3\times 9K}$$




$X_i$: position $r_i$: rotation $s_i$: scaling $E_i$: expression basis
$a_i$: albedo $o_i$: roughness $S_i$: shape basis $P_i$: pose basis
$\alpha_i$: opacity $W_i$: blend weights $f_{0_i}$: fresnel reflectance

Learnable parameters

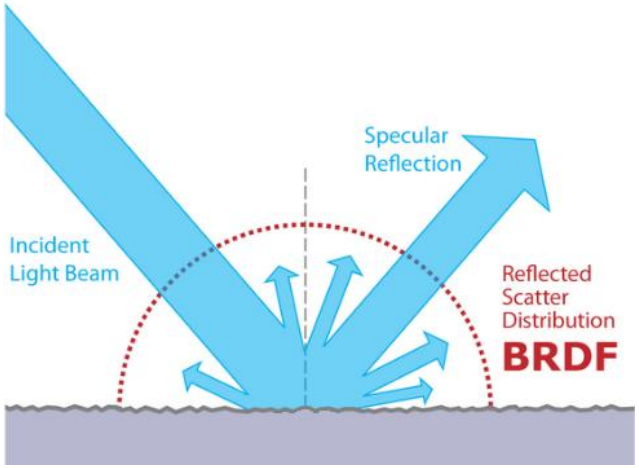
$\mathcal{BS}$: linear blendshapes
$\mathcal{LBS}$: linear blend skinning
$\oplus$: element wise addition

Operations

# Method

- Appearance Modeling
  - Albedo $a$, Roughness $o$, Fresnel base Reflectance $f_o$ 를 이용해서 Appearance Modeling
    - Disney-BRDF 모델을 이용해서 Rendering
      - Bidirectional reflectance distribution function(BRDF)
        - ✓들어온 빛이 표면에서 어느 방향으로 얼마나 튕겨 나가는지를 알려주는 반사 분포

- Albedo map A $\in \mathbb{R}^3$
- Roughness O $\in [0,1]$, O $\in [\tau^o_{min}, \tau^o_{max}]$
- Fresnel base Reflectance $F_o \in \mathbb{R}^3$, $F_o \in [\tau^{f_o}_{min}, \tau^{f_o}_{max}]$
- Normal map N
- Viewing direction V
- Reflection direction map $\mathrm{R} = 2(\mathrm{N} \cdot \mathrm{V})\mathrm{N} - \mathrm{V}$
- $\mathrm{I}_{shading} = \mathrm{I}_{diffuse} + \mathrm{I}_{specular}$



# Method

- Appearance Modeling
  - Diffuse BRDF (난반사)

$$- I_{diffuse}(x) = \int_{\Omega} f_d(\omega_i) L_i(x, \omega_i)(n \cdot \omega_i) + d\omega_i = \frac{A(x)}{\pi} \int_{\Omega} L_i(x, \omega_i)(n \cdot \omega_i) + d\omega_i = \frac{A(x)}{\pi} E_{irr}(n)$$

$$- I_{diffuse} = A \cdot I_{irr}(\mathrm{N}), \quad I_{irr}(\mathrm{N}) \approx \frac{1}{\pi} \int_{\Omega} L_i(\omega)(\mathrm{N} \cdot \omega) + d\omega$$

  - Specular BRDF (정반사)

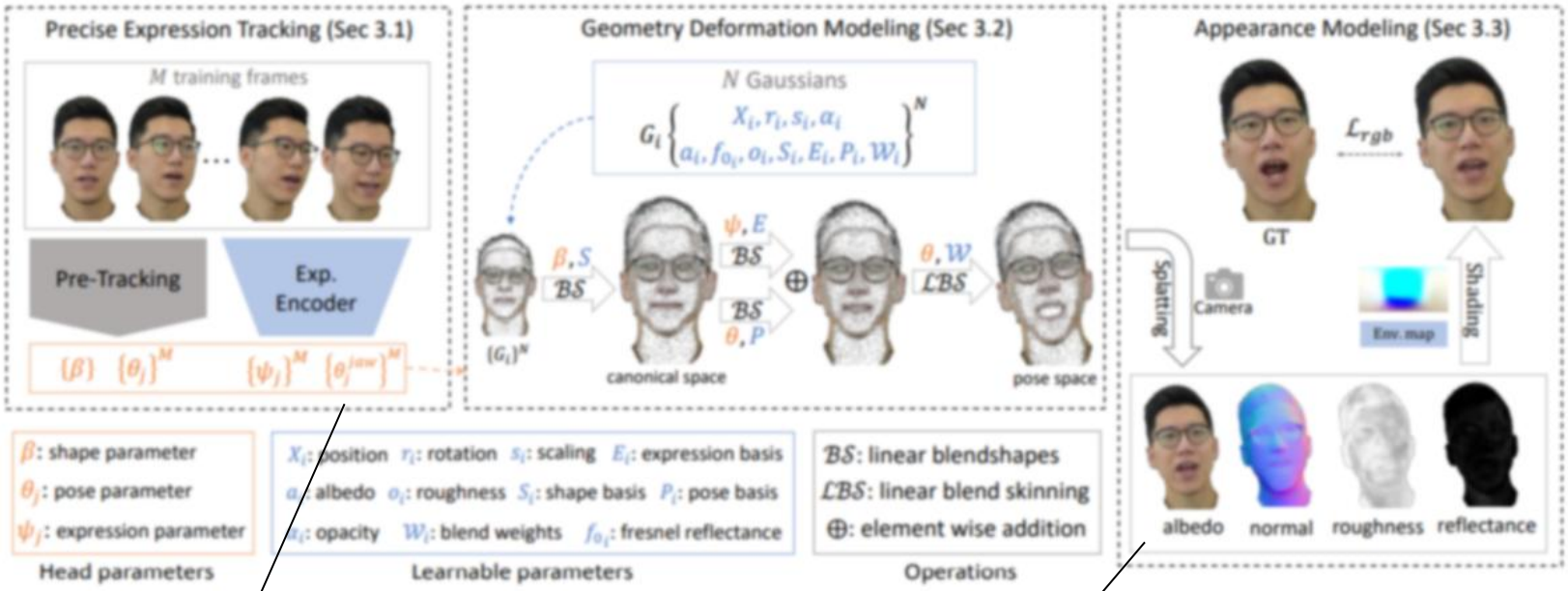
$$- I_{specular} = \int_{\Omega} f_s(\omega_i, \omega_o) L_i(\omega_i)(n \cdot \omega_i) d\omega_i, \quad f_s(\omega_i, \omega_o) = \frac{DFG}{4(n \cdot \omega_i)(n \cdot \omega_o)}$$

$$- I_{specular} \approx I_{env}(\mathrm{R}, \mathrm{O}) \cdot (k_s \cdot I_{BRDF}(\mathrm{O}, \mathrm{N} \cdot \mathrm{V})[0] + I_{BRDF}(\mathrm{O}, \mathrm{N} \cdot \mathrm{V})[1])$$

$$\therefore k_s = F_o + (max(1 - O, F_o) - F_o) \cdot 2^{(-5.55473(N \cdot V - 6.698316) \cdot (N \cdot V))}$$

# Method

- Loss Function
  - $L_{total} = L_{rgb} + \lambda_{jaw}L_{jaw} + \lambda_{normal}L_{normal} + \lambda_{albedo}L_{albedo} + \lambda_{tv}L_{tv}(O)$
    - $L_{rgb} = \lambda_1 ||I_{shading} - I_{gt}||_1 + (1-\lambda_1)L_{D-SSIM}(I_{shading}, I_{gt})$




$$L_{jaw} = ||\hat{\theta}^{jaw} - \theta^{jaw}||_2$$

$$L_{normal} = ||1 - N \cdot \hat{N}||_1$$

$$L_{albedo} = ||A - A^{gt}||_1$$

# Method

- Experiment
  - INSTA(IN-the-wild Speech-driven Talking Avatars)
    - 실제 유튜브/인터넷 영상에서 수집된 다양한 화자 얼굴 영상
    - 10 Subject를 사용, 각 Subject의 마지막 350 프레임을 self-reenactment test-set으로 사용
  - HDTF(High-Definition Talking Face Dataset)
    - 영화, 드라마, 고화질 유튜브 영상 등에서 추출한 고해상도 얼굴 영상
    - 8 Subject를 사용, 마지막 500 frame을 test-set으로 사용
  - Self-Captured Dataset
    - 휴대폰으로 촬영, 5 Subject
    - 마지막 500 frame을 test-set으로 사용
- Baseline
  - Point-avatar, INSTA, Splatting-avatar, Flash-avatar, 3D Gaussian Blendshapes, FLARE
- Metric
  - PSNR, SSIM, LPIPS, MAE

# Method

- Experiment
  - Quantitative Results

| Method | INSTA dataset | | | | HDTF dataset | | | | self-captured dataset | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | MAE*↓ | SSIM↑ | LPIPS↓ | PSNR↑ | MAE*↓ | SSIM↑ | LPIPS↓ | PSNR↑ | MAE*↓ | SSIM↑ | LPIPS↓ |
| INSTA | 27.85 | 1.309 | 0.9110 | 0.1047 | 25.03 | 2.333 | 0.8475 | 0.1614 | 25.91 | 1.910 | 0.8333 | 0.1833 |
| Point-avatar | 26.84 | 1.549 | 0.8970 | 0.0926 | 25.14 | 2.236 | 0.8385 | 0.1278 | 25.83 | 1.692 | 0.8556 | 0.1241 |
| Splatting-avatar | 28.71 | 1.200 | 0.9271 | 0.0862 | 26.66 | 2.01 | 0.8611 | 0.1351 | 26.47 | 1.711 | 0.8588 | 0.1550 |
| Flash-avatar | 29.13 | 1.133 | 0.9255 | 0.0719 | 27.58 | 1.751 | 0.8664 | 0.1095 | 27.46 | 1.632 | 0.8348 | 0.1456 |
| GBS | 29.64 | 1.020 | 0.9394 | 0.0823 | 27.81 | 1.601 | 0.8915 | 0.1297 | 28.59 | 1.331 | 0.8891 | 0.1560 |
| HRAvatar (Ours) | **30.36** | **0.845** | **0.9482** | **0.0569** | **28.55** | **1.373** | **0.9089** | **0.0825** | **28.97** | **1.123** | **0.9054** | **0.1059** |

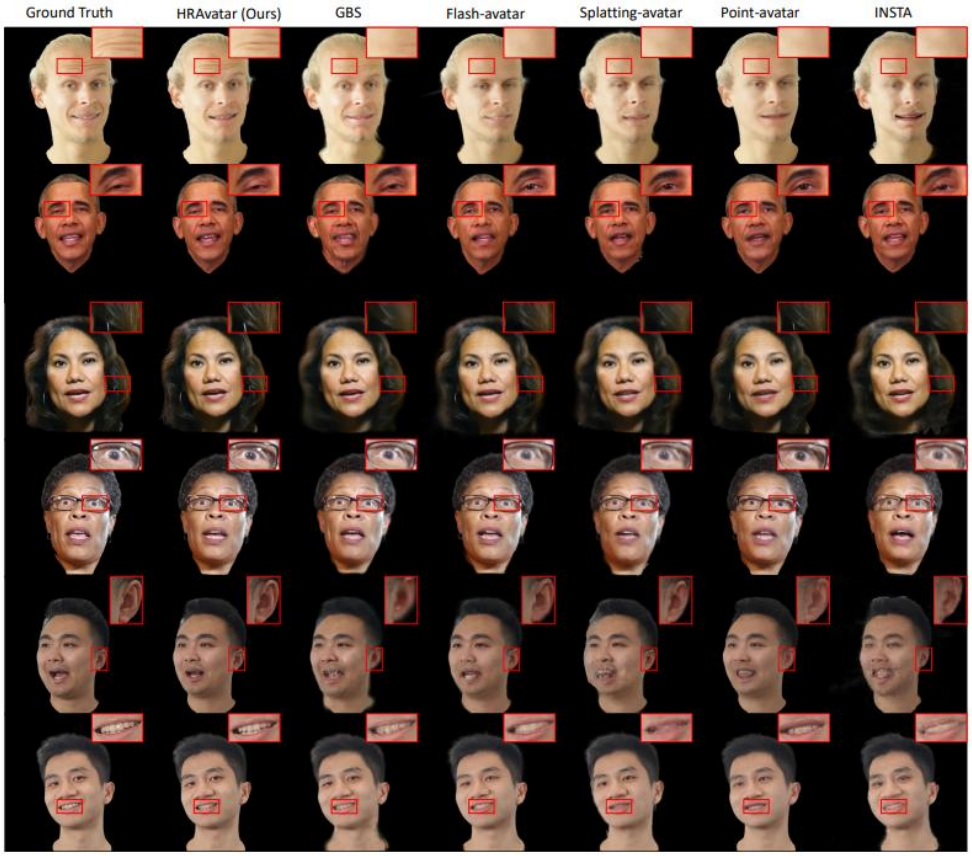
| Method | INSTA dataset | | | | HDTF dataset | | | | self-captured dataset | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | MAE*↓ | SSIM↑ | LPIPS↓ | PSNR↑ | MAE*↓ | SSIM↑ | LPIPS↓ | PSNR↑ | MAE*↓ | SSIM↑ | LPIPS↓ |
| FLARE | 26.80 | 1.433 | 0.9063 | 0.0816 | 25.55 | 2.193 | 0.8479 | 0.1183 | 25.82 | 1.715 | 0.8576 | 0.1230 |
| HRAvatar (Ours) | **30.36** | **0.845** | **0.9482** | **0.0569** | **28.55** | **1.373** | **0.9089** | **0.0825** | **28.97** | **1.123** | **0.9054** | **0.1059** |

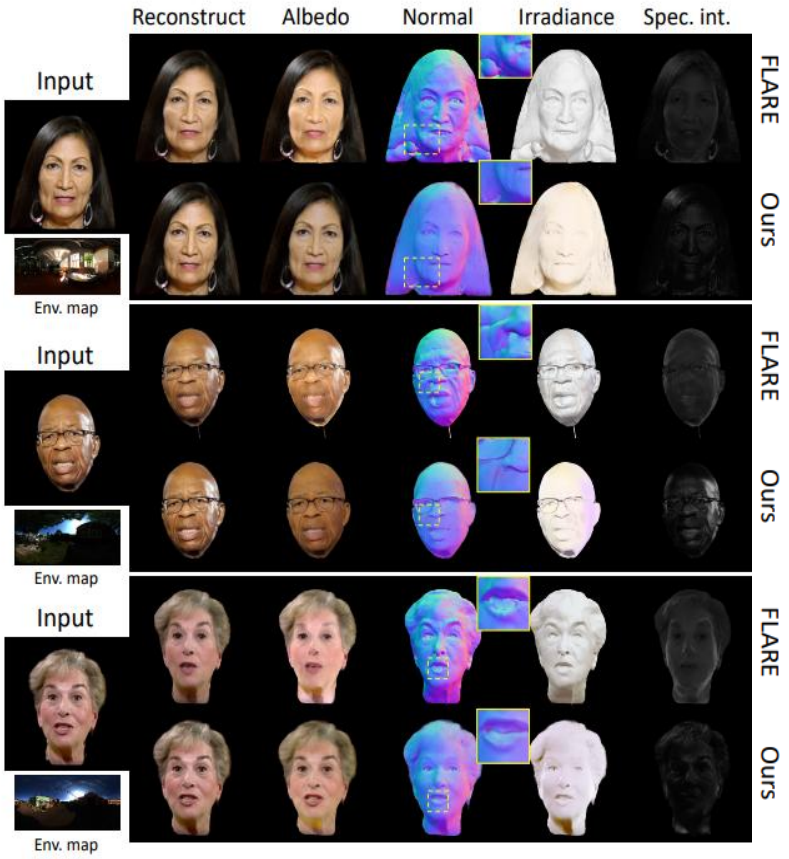
| | Rendering Quality | | Relighting | Rendering speed |
|---|---|---|---|---|
| Point-Avatar [80] | | 0.646 | Limited | ≈ 6 FPS |
| INSTA [82] | | 0.764 | ✗ | ≈ 1 FPS |
| FLARE [2] | | 0.698 | ✓ | ≈ 35 FPS |
| Splatting-avatar [57] | | 0.834 | ✗ | > 120 FPS |
| Flash-avatar [62] | | 0.883 | ✗ | > 120 FPS |
| GBS [46] | | 0.980 | ✗ | > 120 FPS |
| HRAvatar (Ours) | | 1.184 | ✓ | > 120 FPS |

Table 3. Key aspects of our method compared to previous works. The rendering quality shows the inverse of the MAE metric on the INSTA dataset, with longer bars representing better performance. 'Limited' indicates that the Point-Avatar method has limited flexibility in handling relighting.

# Method

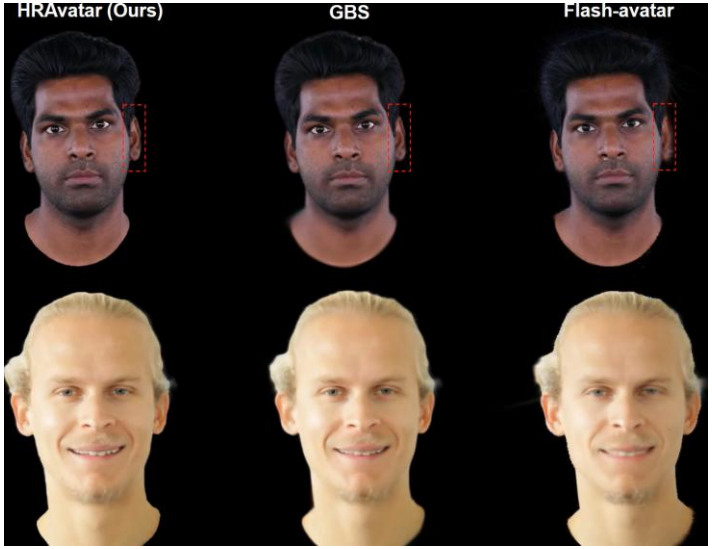
- Experiment
  - Qualitative Results



# Method

- Experiment




**Self-reenactment**


**Relighting**


**Cross-reenactment**

# 감사합니다.