MoAI 세미나

# Dexterous Manipulation

잘 보고, 잘 느끼는 로봇 손: 3D Gaussian Splatting에서 촉각까지

잘 보기 · 시각 잘 느끼기 · 촉각

윤서현 · 2026년 11월 3일

발표 흐름

# 정교한 조작을 잘하려면 두 가지가 필요합니다

1부 · 잘 보기

### 3D Gaussian Splatting

물체와 장면을 3D로 빠르고 정확하게 복원

2부 · 보기의 활용

### ViserDex

3DGS로 로봇 손의 물체 돌리기를 학습

3부 · 잘 느끼기

### T-Rex

손끝 촉각으로 실시간 힘 조절

4부 · 합치기

### Touch-GS

촉각으로 3DGS의 빈틈을 보완

앞 발표: **정교한 조작이란?** → 이번 발표: **어떻게 하면 잘할 수 있을까?**

# 사진 몇 장으로 3D 장면 만들기

**입력**

**여러 각도에서 찍은 사진**

+ 각 사진의 카메라 위치

→

**복원**

**3D 장면 표현 만들기**

사진들과 맞도록 학습

→

**출력**

**새로운 각도에서 렌더링**

찍지 않은 시점의 모습

이 문제를 **새로운 시점 합성(Novel View Synthesis)**이라고 합니다.

로봇에게는 → **물체의 3D 모양과 위치를 정확히 아는 것**

# 기존 방법 NeRF: 예쁘지만 너무 느리다

### NeRF는 이렇게 그립니다

- 장면을 신경망(MLP) 안에 저장
- 픽셀마다 광선을 따라가며 신경망에 수백 번 질문
- 아무것도 없는 빈 공간까지 계산

**최대 48시간**

Mip-NeRF360의 학습 시간

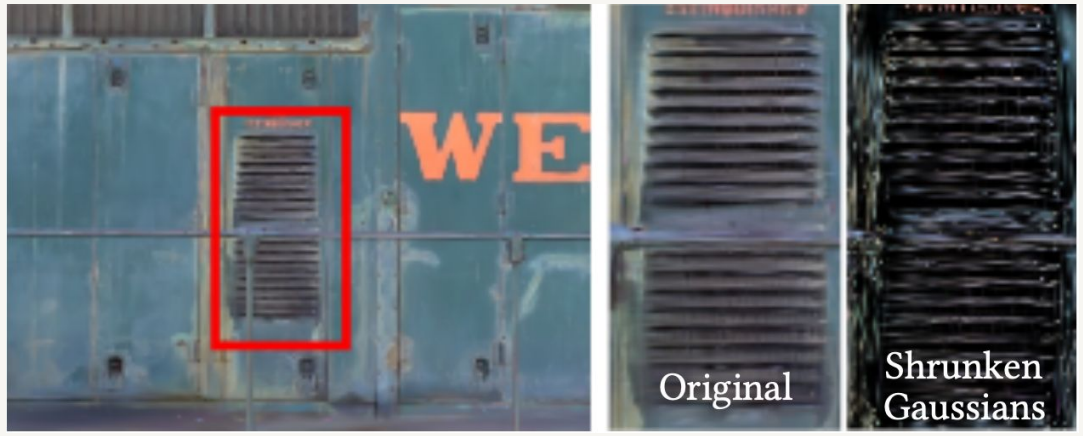
**0.06 fps**

1초에 0.06장 → 한 장 그리는 데 약 17초

# 3DGS의 아이디어: 흐릿한 색 구슬로 칠하기

## 장면을 반투명한 3D 가우시안 수백만 개로 표현한다

- 가우시안 = 가장자리가 흐릿한 타원체 모양 덩어리
- 점묘화처럼, 많이 모이면 그럴듯한 그림이 된다
- 신경망 없이 구슬만으로 장면을 직접 표현




왼쪽: 렌더링 결과

오른쪽: 가우시안을 60% 축소해 모양을 보여줌

# 구슬 하나가 가진 네 가지 정보

### 위치 μ
공간의 어디에 있는가

### 모양과 방향 Σ
동그란지, 납작한지, 길쭉한지 · 크기 S와 회전 R로 나눠 학습

### 불투명도 α
얼마나 비치는가

### 색 SH 계수
보는 각도에 따라 달라지는 색까지 표현

학습이 끝나면 장면 하나에 구슬 **100만~500만 개**

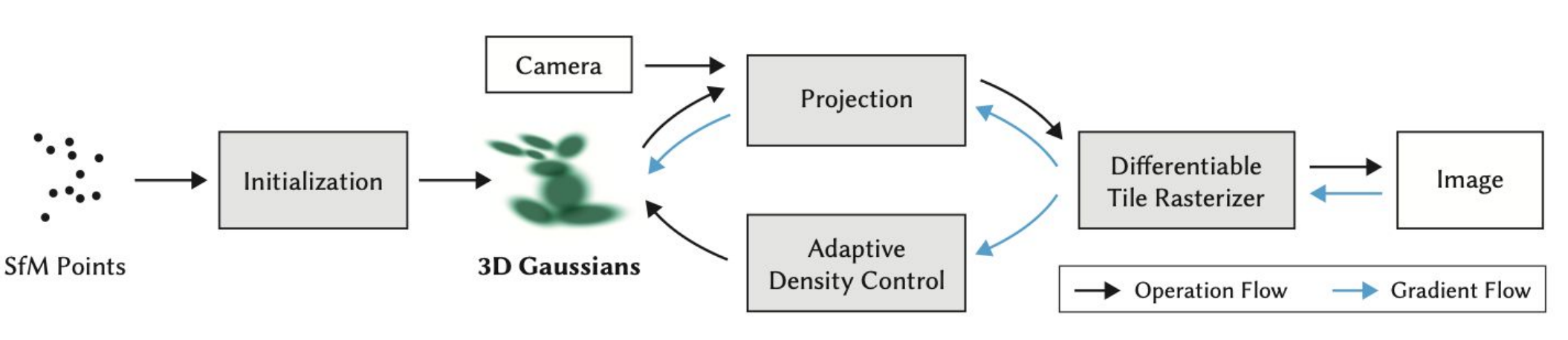
# 전체 과정 한눈에 보기

**STEP 1**
### SfM 점 구름
사진 분석으로 얻은 듬성듬성한 점

**STEP 2**
### 가우시안 초기화
점마다 구슬을 하나씩 올림

**STEP 3**
### 최적화 + 밀도 조절
비교하고 고치기를 반복

**STEP 4**
### 타일 기반 렌더링
빠르게 화면에 그리기



# 학습: 그려보고, 비교하고, 고치기

**① 렌더링**
지금의 구슬들로 사진과 같은 각도에서 그려보기

→

**② 비교**
정답 사진과 얼마나 다른지 계산하기

→

**③ 수정**
위치, 모양, 색, 투명도를 조금씩 조정하기

**비교하는 기준 (손실 함수)**

**0.8 × L1 + 0.2 × D-SSIM**

픽셀 차이와 구조적 차이를 함께 본다

**반복 횟수**

**약 30,000번** 반복

(7,000번만 해도 쓸 만한 결과)

# 구슬 늘리기와 줄이기 (적응형 밀도 제어)

**복제 Clone · 부족한 곳**

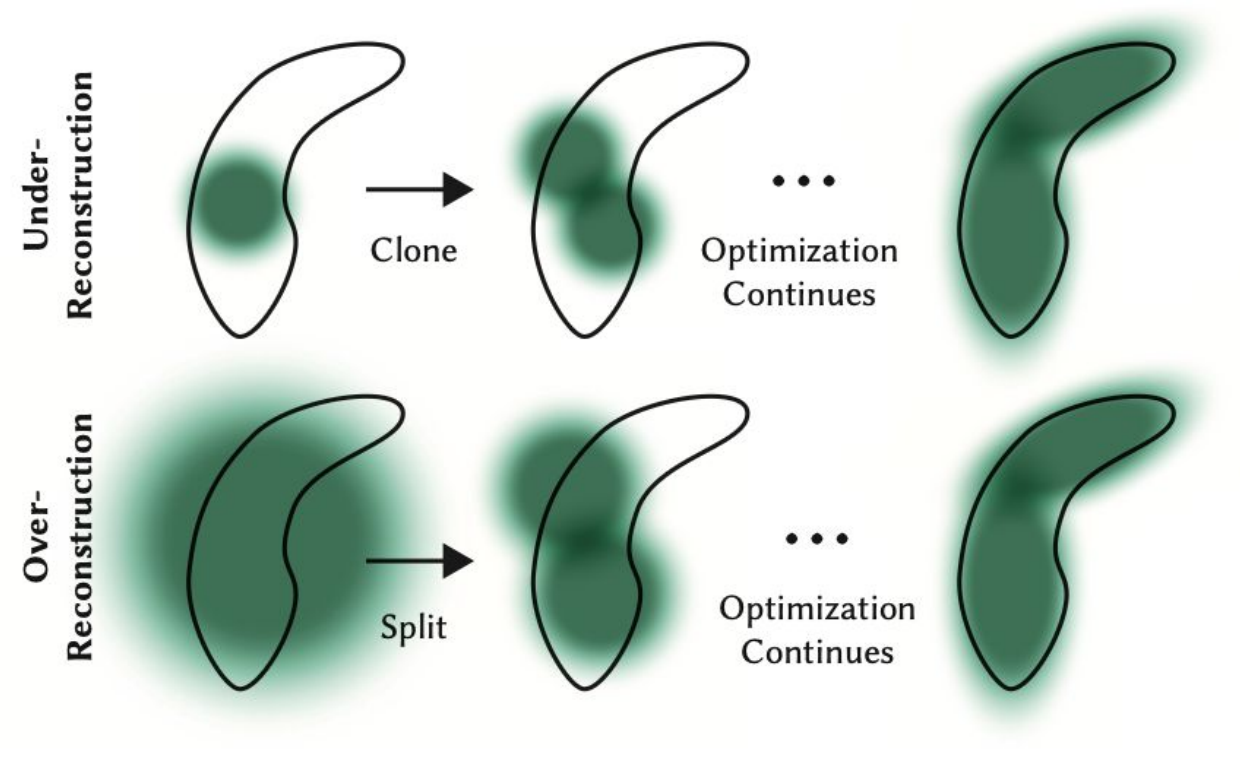
구슬이 작아 다 못 덮으면 → 복사해서 옆에 하나 더

**분할 Split · 뭉개진 곳**

큰 구슬 하나가 넓게 덮으면 → 작은 구슬 두 개로

**제거 Prune · 쓸모없는 곳**

거의 투명한 구슬은 지우기



# 빠른 렌더링: 구슬을 화면에 찍는다

**NeRF** · 픽셀마다 신경망에 수백 번 질문

**3DGS** · 구슬을 화면에 바로 찍기 (splat)

STEP 1

## 타일 나누기

화면을 16×16 픽셀 칸으로 나눠 동시에 처리

STEP 2

## 한 번에 정렬

모든 구슬을 칸과 깊이 순서로 GPU에서 정렬

STEP 3

## 앞에서부터 쌓기

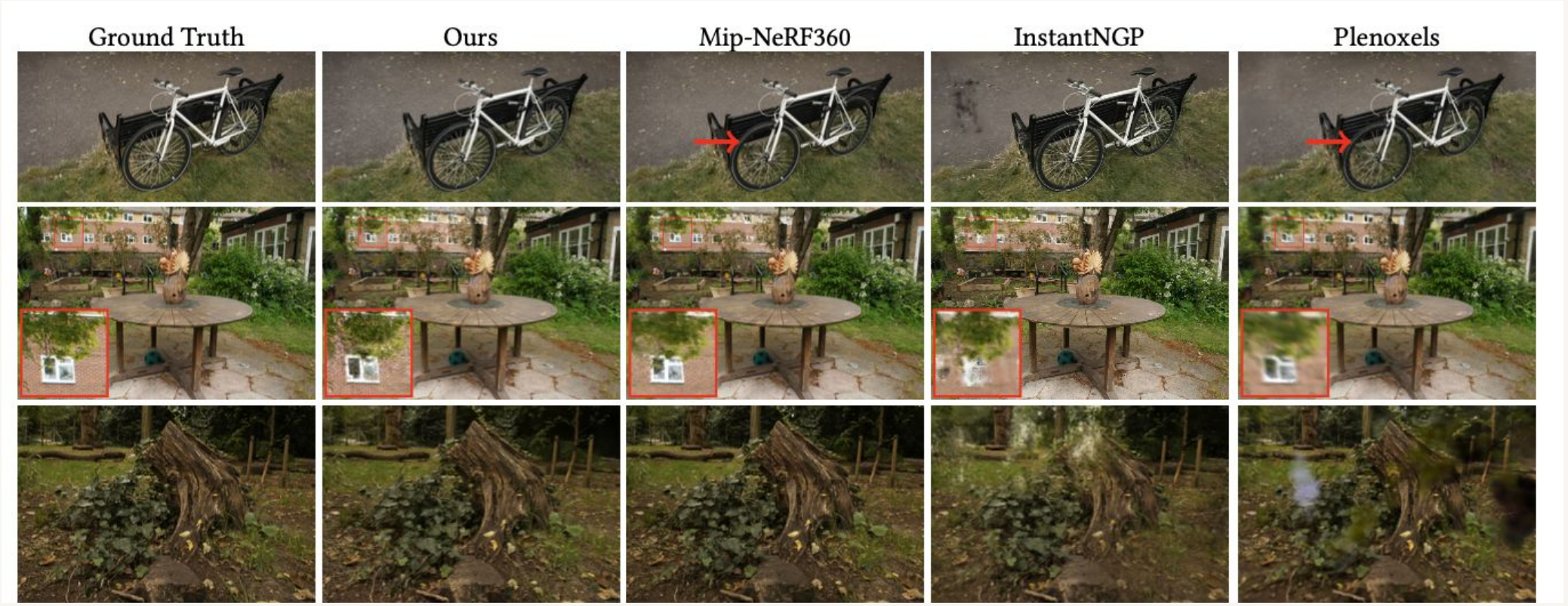
셀로판지처럼 색을 겹치고, 꽉 차면 멈춤

# 결과: 화질은 그대로, 속도는 수백 배

| 항목 | Mip-NeRF360 | 3DGS |
|---|---|---|
| 학습 시간 | 최대 48시간 | **약 41분** |
| 속도 | 0.06 fps | **134 fps** |
| 화질 | 최고 수준 | **비슷하거나 더 좋음** |
| 용량 | 작음 | 수백 MB |

134 fps = 1초에 134장 → 게임처럼 **실시간**으로 돌려볼 수 있다

# 눈으로 비교하기: 방법별 화질



# 한계: 안 보이는 곳은 만들 수 없다

### 큰 용량

구슬이 수백만 개라 장면 하나에 수백 MB

### 덜 찍힌 영역

사진이 부족한 곳은 얼룩지거나 뭉개짐 (Fig. 11, 12)

### 튀는 현상

시점을 돌릴 때 구슬이 갑자기 나타나거나 사라짐

로봇 손의 입장에서 보면

**손으로 물체를 쥐는 순간, 가려진 부분은 카메라로 볼 수 없다**

2부 · 보기의 활용 — ViserDex

# 물체를 돌리려면 자세를 알아야 한다

## 과제: 손 안에서 물체를 원하는 방향으로 돌리기 (In-hand reorientation)

### 필요한 것

물체가 지금 어떤 방향으로 놓였는지, 정확한 자세(pose) 추정

### 기존의 한계

카메라 여러 대가 필요하거나, 비싼 레이 트레이싱 렌더링에 의존

**ViserDex**: Visual Sim-to-Real for Robust Dexterous In-hand Reorientation · Bhardwaj et al. · ETH Zürich · RSS 2026

# 3DGS로 진짜 같은 학습 데이터 만들기

**STEP 1**

## 물체를 3DGS로

실제 물체를 가우시안 구슬로 복원

**STEP 2**

## 무작위로 바꾸기

가우시안에 물리적으로 그럴듯한 변형을 줘 다양한 이미지 생성

**STEP 3**

## 로봇 학습

자세 추정기 + 강화학습 조작 정책 (교사-학생 증류)

**핵심**

시뮬레이션 화면과 실제 카메라 화면의 차이 **(sim-to-real gap)**를 줄인다

# 결과: 카메라 하나로 물체 돌리기 성공

### 실제 로봇에서 검증

다지 로봇 손 + RGB 카메라로 물체 5종 재배향 성공, 까다로운 조명에서도 안정적

### 3DGS 데이터의 효과

3DGS 데이터로 학습한 자세 추정기가 기존 렌더링 데이터보다 우수

하지만

**손으로 쥐는 순간 물체는 가려진다. 사람은 이때 촉감을 쓴다**

3부: 잘 느끼기 →

# 로봇 AI는 왜 촉각을 잘 못 쓸까

기존 **VLA 모델**(보고, 말을 이해하고, 행동하는 로봇 AI)은 대부분 카메라만 본다. 촉각을 써도 **힘이 시간에 따라 변하는 모습**은 놓친다.

### 데이터 부족

다양한 촉각 데이터와 표준 평가가 없다

### 구조의 한계

VLA 구조가 빠른 촉각 반응에 맞지 않다

### 약한 인코더

촉각 신호를 제대로 해석하지 못한다

**T-Rex**: Tactile-Reactive Dexterous Manipulation · Niu, Liu, Wang et al. · UC Berkeley, NVIDIA, Stanford 외 · arXiv 2606.17055 (2026)

# 하드웨어: 촉각을 가진 두 손 로봇

몸체

## Dexmate Vega-1 양팔 로봇

7자유도 팔 × 2

손

## Sharpa Wave 로봇 손

22자유도 손 × 2

촉각

## 손가락 끝 센서 5개 / 손

눌린 모양(변형 맵) + 6축 힘·토크

시각

## 카메라 3대

머리 스테레오 카메라 + 손목 카메라 2대

# 데이터: 촉각까지 함께 기록한 100시간

**100시간**
사람이 원격 조종한 시연 데이터

**207개**
가정에서 쓰는 다양한 물체

**22가지**
기본 동작 단위 (motor primitive)

촉각 신호를 영상·동작과 **시간을 맞춰** 기록하고, 데이터셋을 **공개**했다

# 방법 ①: 느린 뇌와 빠른 반사신경

느린 뇌 · 행동 전문가 (Action expert)

## 카메라를 보고 큰 동작 계획

예: “컵 쪽으로 손을 뻗자”

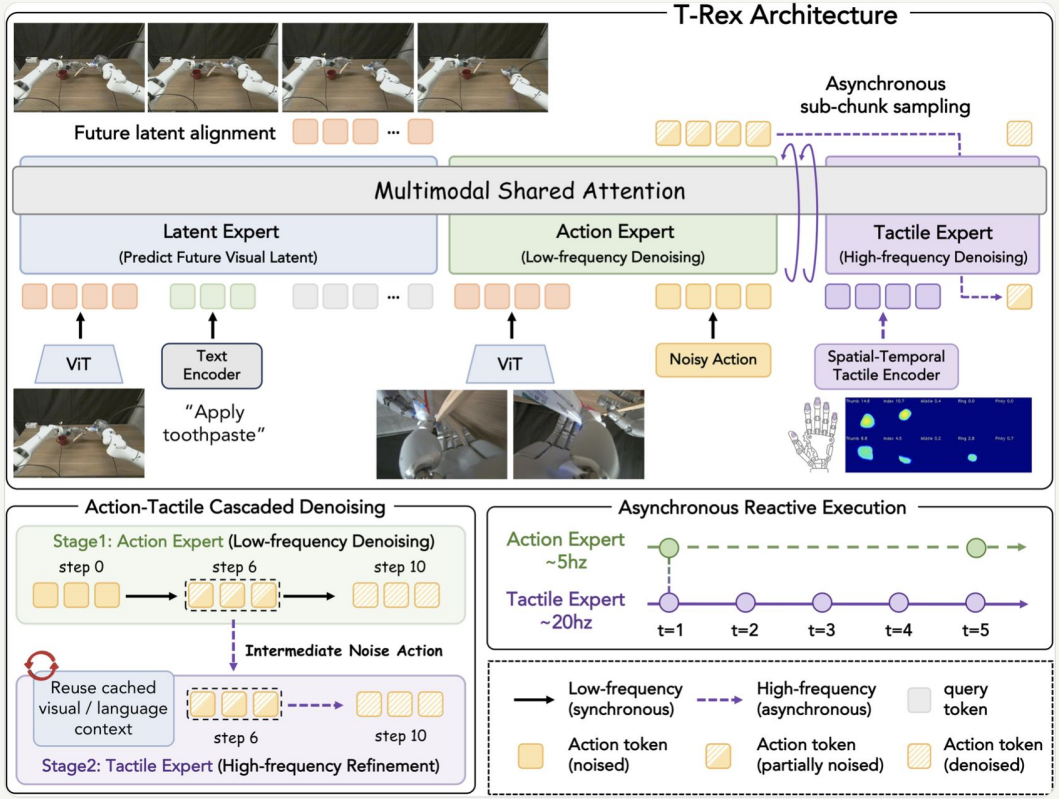
느린 주기로 실행

빠른 반사신경 · 촉각 전문가 (Tactile expert)

## 손끝 힘을 느끼고 미세 보정

예: “미끄러진다, 살짝 더 꽉!”

빠른 주기로 실행



# 방법 ②: 촉각을 단어로 바꾸고 3단계로 학습

촉각 토큰화

## 힘 신호 → AI가 읽는 “단어”

짧은 시간 동안의 손끝 힘 데이터를 **Temporal VQ-VAE**로 압축해, 몇 개의 이산 토큰으로 바꾼다

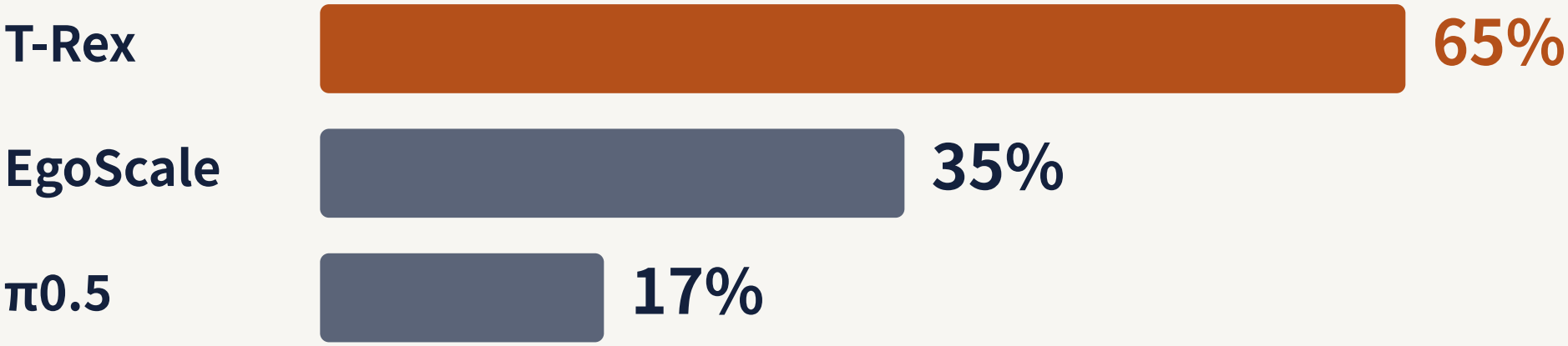
**1단계 · 사전학습**
사람 1인칭 영상 약 22,900시간

**2단계 · 중간학습**
100시간 로봇 촉각 데이터

**3단계 · 미세조정**
작업마다 시연 약 100개

# 결과: 실제 작업 12개에서 최고 성능

평균 성공률




작업당 16번 시도 · **12개 작업 모두에서** 가장 높은 성공률

# 촉각을 빼면 어떻게 될까?

**65%** → **42%**

촉각 있음 / 촉각 제거

**성공률 23%p 하락 → 정교한 조작에서 촉각은 필수**

# 3DGS가 약한 물체: 거울, 투명, 반짝임

## 사진만으로는 표면이 정확히 어디에 있는지 알기 어렵다

### 거울·반사
비친 모습을 진짜 물체로 착각

### 투명한 물체
뒤가 비쳐 표면이 보이지 않음

### 적은 사진
시점이 적으면 모양이 부정확

**Touch-GS**: Visual-Tactile Supervised 3D Gaussian Splatting · Swann, Strong et al. · Stanford · IROS 2024

# 방법: 만져서 얻은 깊이로 3DGS를 가르친다

**STEP 1**
### 여러 번 터치
DenseTact 촉각 센서를 단 Kinova Gen3 로봇 팔

→

**STEP 2**
### 표면 추정
GPIS로 터치들을 합쳐 표면과 불확실성 계산

→

**STEP 3**
### 깊이 맵 합치기
단안 깊이 추정을 깊이 카메라, 촉각 순으로 맞춤

→

**STEP 4**
### 3DGS 학습
불확실성 가중 깊이 손실을 추가

**촉각**이 정확한 깊이를 알려주고, **시각**이 넓은 범위를 채운다

# 결과: 보기의 빈틈을 느끼기로 채운다

시각만

### 어려운 물체에서 실패

거울·투명 물체의 모양을 잘못 복원

촉각만

### 범위가 좁음

만진 곳 주변만 알 수 있음

시각 + 촉각

### 가장 좋은 결과

정량·정성 평가 모두 우수, 적은 시점과 거울·투명 물체에서도

→ **로봇 손이 물체를 만지는 순간**, 그 정보로 3D 이해를 더 정확하게 만들 수 있다

정리

# 잘 보고 + 잘 느끼는 로봇 손

| 연구 | 역할 | 핵심 |
|---|---|---|
| **3DGS** | 잘 보기 (기초) | 구슬로 3D를 빠르고 정확하게 |
| **ViserDex** | 보기 → 조작 | 3DGS로 로봇 손 학습 데이터 |
| **T-Rex** | 잘 느끼기 | 촉각 반사신경, 성공률 65% |
| **Touch-GS** | 보기 + 느끼기 | 촉각으로 3DGS 빈틈 보완 |

**앞으로: 3DGS의 3D 이해 + T-Rex의 촉각 반응 → 더 사람 같은 로봇 손**

# 감사합니다

**참고 문헌**

Kerbl et al., “3D Gaussian Splatting for Real-Time Radiance Field Rendering,” ACM TOG (SIGGRAPH), 2023.

Bhardwaj et al., “ViserDex: Visual Sim-to-Real for Robust Dexterous In-hand Reorientation,” RSS, 2026.

Niu, Liu, Wang et al., “T-Rex: Tactile-Reactive Dexterous Manipulation,” arXiv:2606.17055, 2026.

Swann, Strong et al., “Touch-GS: Visual-Tactile Supervised 3D Gaussian Splatting,” IROS, 2024.