Мультимодальные и поведенческие модели

Teleoperation, Data Collection, and Scaling

MIPT VLA 2026 | 5 октября 2026

Алексей Староверов



МФТИ

Center for Cognitive Modeling

План лекции

Robot Learning Data Collection Methods (UMI)

Simulation & Data Collection

Why Data Collection?

- Current paradigm is imitation learning $\max_{\theta} \mathbb{E}_{(s,a)\sim D}[\log \pi_\theta(a|s)]$
- For LLMs you have the internet (wikipedia, books, etc.)
- Where do we get this for robotics data?
  - We're data poor in the robotics world
- **Sidenote**: You can get away without data collection by doing RL from scratch, but super inefficient (purely learning by reward signal and through exploration)
- Scaling laws?

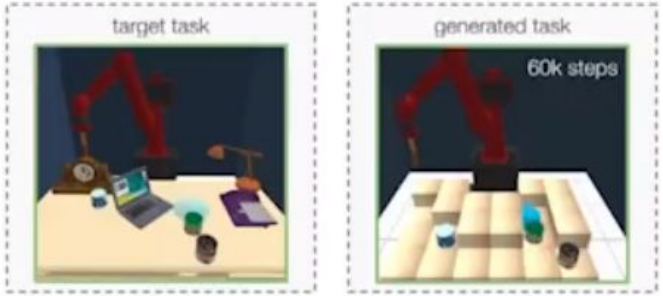
Data Collection Methods pre-~2010s



SpaceMouse /Phantom
Omni Human Joystick



Kinesthetic Teaching



Procedural Generation

Current Data Collection Methods (teleoperation paradigm)



ALOHA (2023)



GELLO (2024)



VR Teleoperation (source: Astribot)

Teleoperation for robot hands (very similar)



Manus Gloves



VR Teleop

Problem: Price + slow (due to operator embodiment-gap / latency)

Non-teleoperation techniques?



UMI (2024)



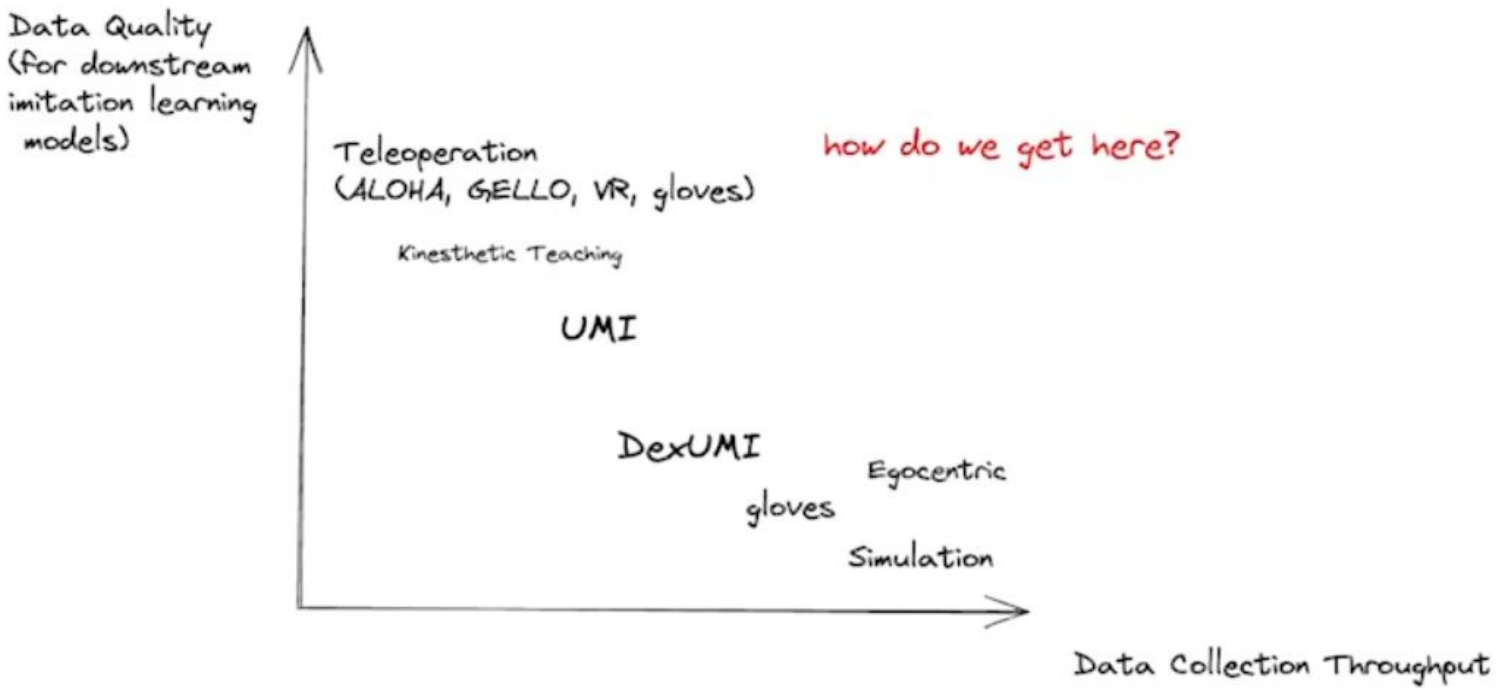
DexCap (2025)



EgoDex (2025)

Good for occlusions

Data collection Methods



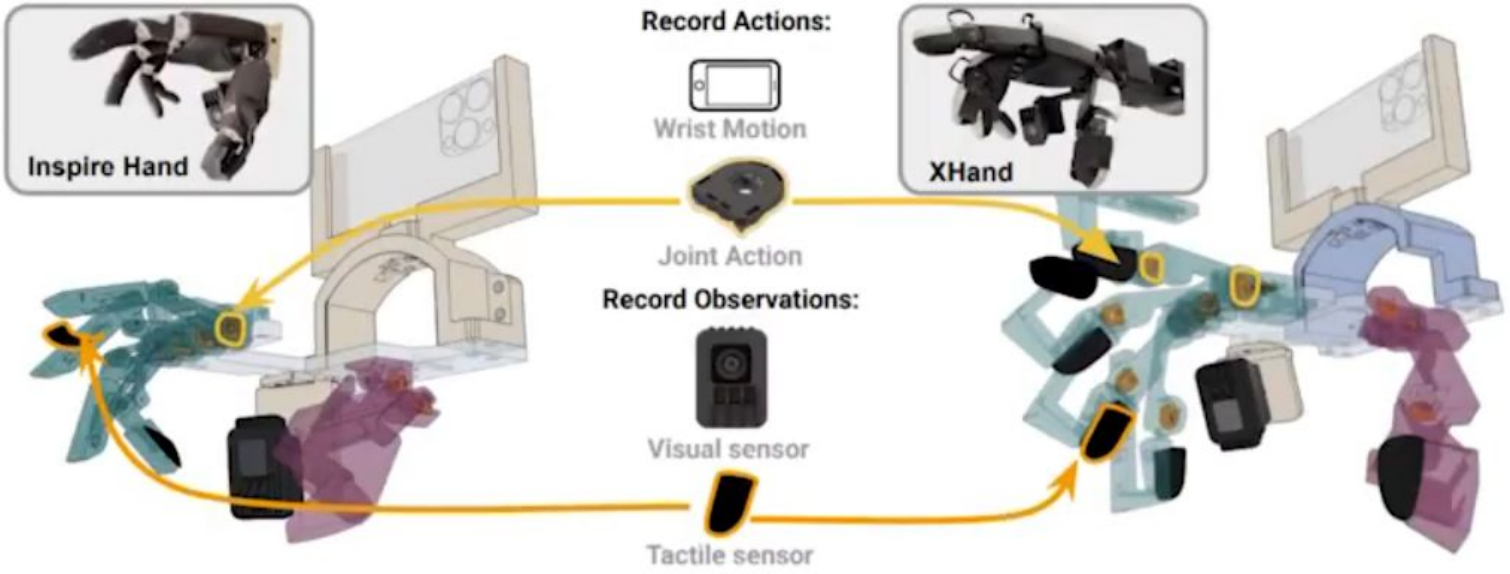
The problem with egocentric data

- **Kinematic / embodiment gap** (differences in robot vs human arm structure, degrees of freedom, hands vs. parallel grippers) → physically impossible to replicate same motion

- **Visual / appearance gap** (differences in how the robot's arms appear vs human hands in images)→ perception system might not recognize as the same thing

- **Dynamics gap** (differences in mass, friction, compliance, and control dynamics between human and robot) → the force applied to each muscle vs. to each motor is completely different

DexUMI


Inspire Hand
Record Actions:
Wrist Motion
XHand
Joint Action
Record Observations:
Visual sensor
Tactile sensor

DexUMI

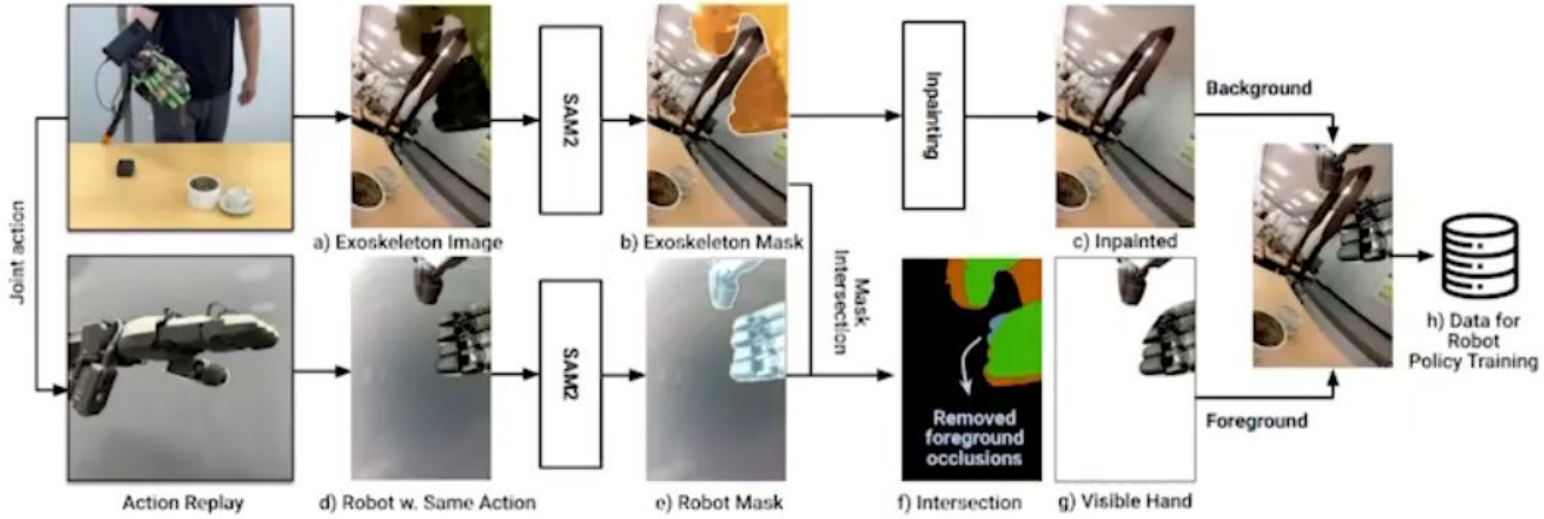


Figure 4: **Bridging the Visual Gap.** To convert the visual observation into policy training data, we first segment the exoskeleton using SAM2 (b) and inpaint the missing background (c). The corresponding joint action (a) is replayed on the dexterous hand to obtain the robot hand image (d). SAM2 is applied to obtain the robot mask (e). The intersection (f) of the exoskeleton mask (b) and robot mask (e) identifies the visible part of the hand during interaction. Finally, we replace pixels in the inpainted background (c) with the visible robot hand (g).

DexUMI

Method			Inspire Hand				XHand				
Action	Tactile	Visual	Cube	Carton	Tea		Tea		Kichen		
					tool	leaf	tool	leaf	knob	pan	salt
Rel	Yes	Inpaint	**1.00**	0.85	**1.00**	0.85	**1.00**	**0.85**	**0.95**	**0.95**	**0.75**
Abs	Yes	Inpaint	0.10	0.35	0.80	0.00	**1.00**	0.25	0.50	0.45	0.00
Rel	No	Inpaint	0.95	**0.90**	**1.00**	**0.90**	0.95	0.80	**0.95**	**0.95**	0.15
Abs	No	Inpaint	0.90	0.85	0.90	0.60	**1.00**	0.75	0.60	0.60	0.0
Rel	No	Mask	0.60	0.10	0.90	0.50	/	/	/	/	/
Rel	No	Raw	0.20	0.05	0.85	0.05	/	/	/	/	/

Table 1: **Evaluation Results.** We report stage-wise accumualted success rate. The experiments compare different combinations of finger action representation (Absolute vs Relative), tactile feedback (Yes vs No), and visual rendering approaches (Inpaint vs Mask/Raw).

Data collection and Policy Training details

E.1 Data Collection

We collected 310 trajectories for Cube Picking task policy training, 175 trajectories for Egg Carton Opening task policy training, and 400 trajectories for Tea Picking Using Tools policy training (for both Inspire Hand and XHand). For the kitchen task, we collected 370 trajectories covering all four sub-tasks, plus an additional 100 trajectories focused solely on knob closing.

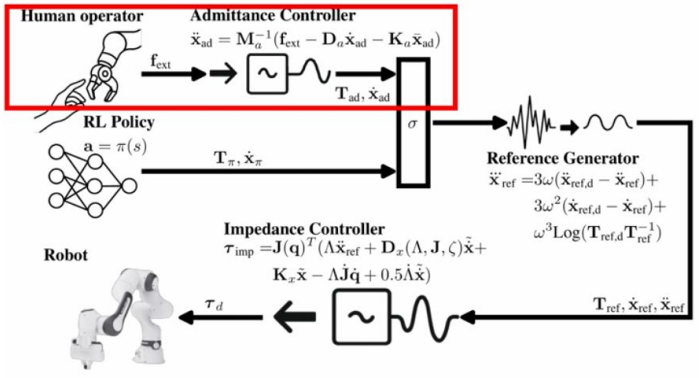
E.3 Policy Training

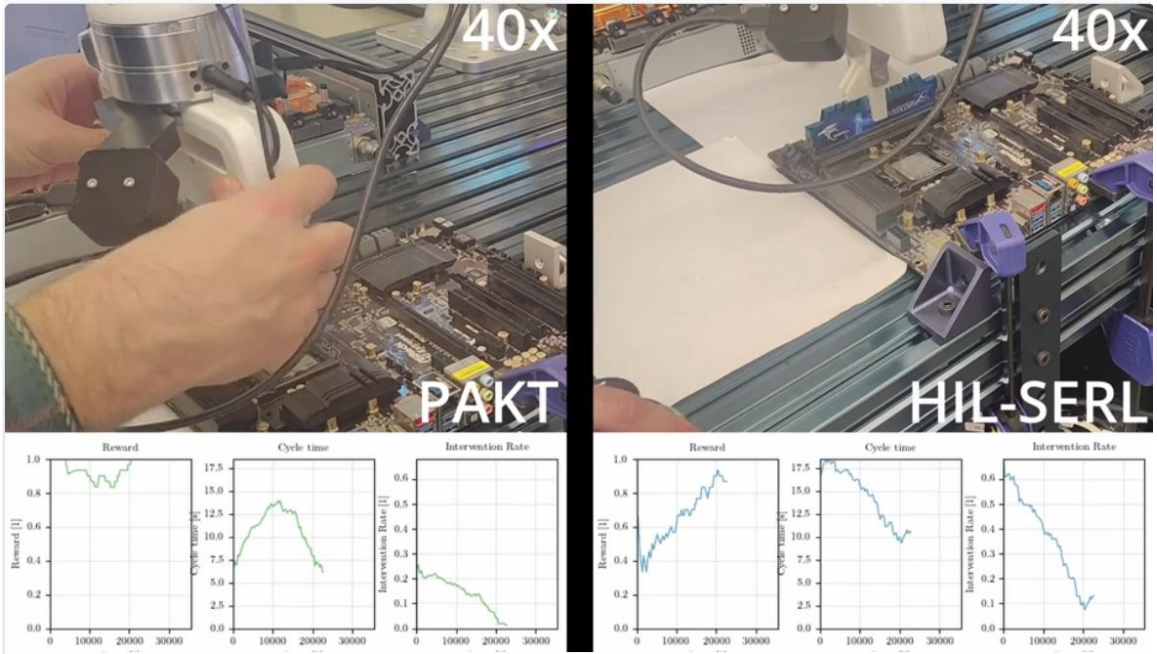
We process the visual observations with pretrained DINO-V2 [76, 77]. Before passing the visual observations into DINO-V2, we augment it with random crop, color jitter, random grayscale and Gaussian Blur. We concatenate the CLS token from DINO-V2 with tactile sensor readings as input to the diffusion policy [78, 79]. The policy predicts 16 steps of robot actions, which contain both 6-DoF robot end-effector relative actions and hand actions (6-DoF for Inspire Hand and 12-DoF for XHand). We train the models for 400 epochs across all tasks for both types of hands. The pretrain DINO-V2 is not frozen and updated during the policy training.

Open Questions

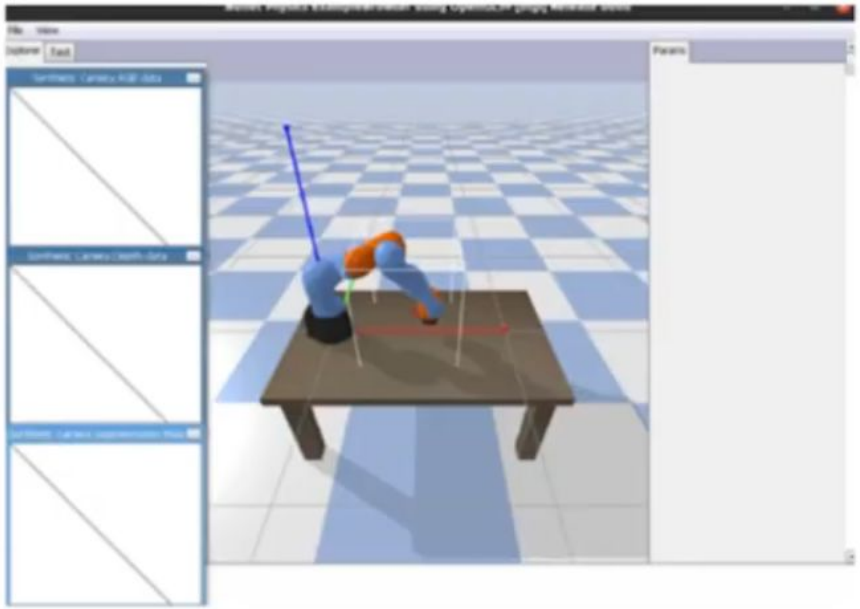
- Limitations of current data collection methods
  - Tactile
  - Active perception
  - Comeback of kinesthetic teaching?
- Can we get away with scaling up data collection?
  - Yes, egocentric data is much easier to scale, but cross-embodiment gap

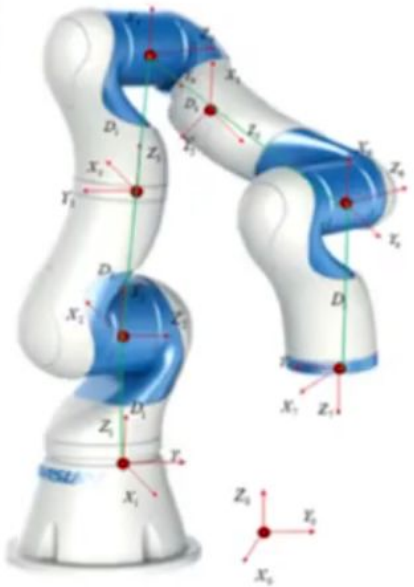
Physically-Aligned Kinesthetic Teaching for Reinforcement Learning



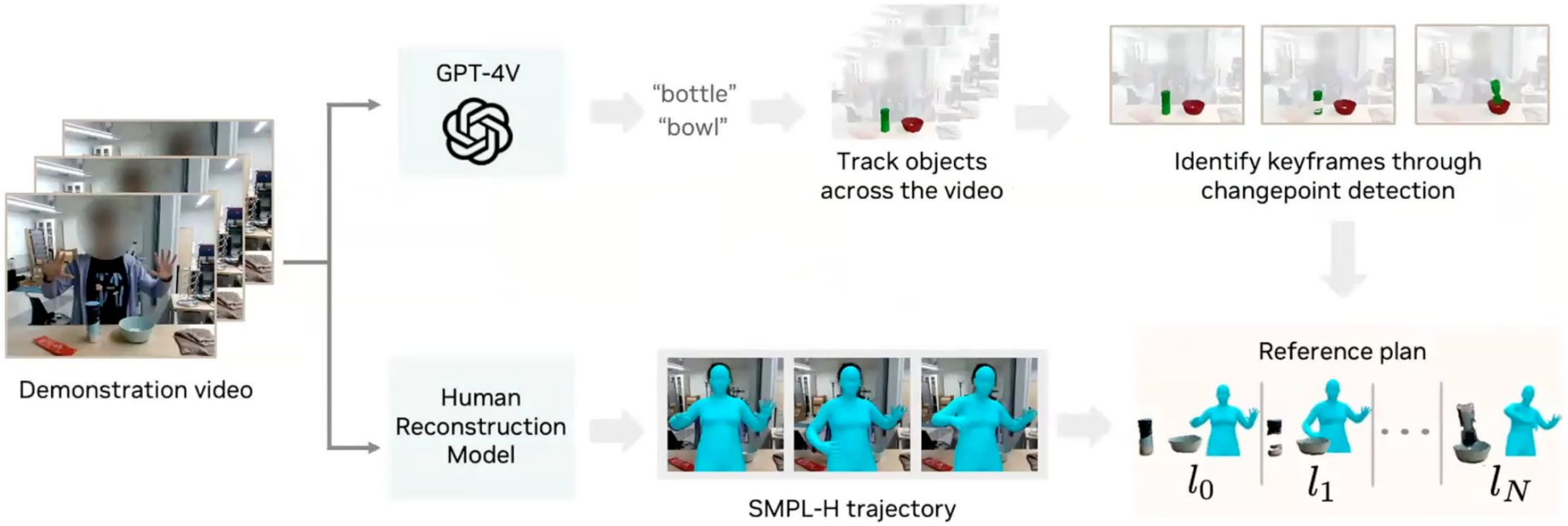


Simulation & Data Collection



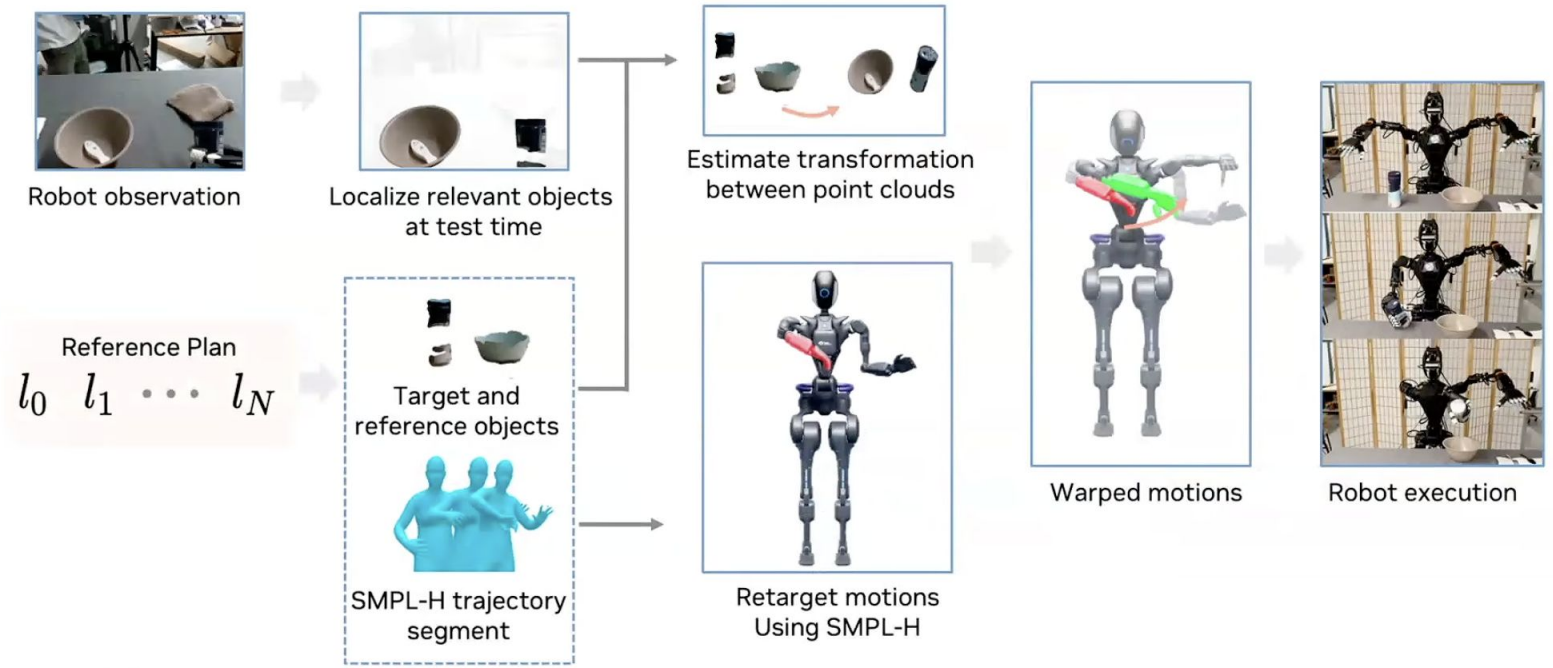


OKAMI: Learning from human videos



In the first stage, OKAMI processes the human video to generate a reference manipulatation plan, a spatiotemporal abstraction of the video that captures which and how objects are moved and how human moves between adjacent subgoals

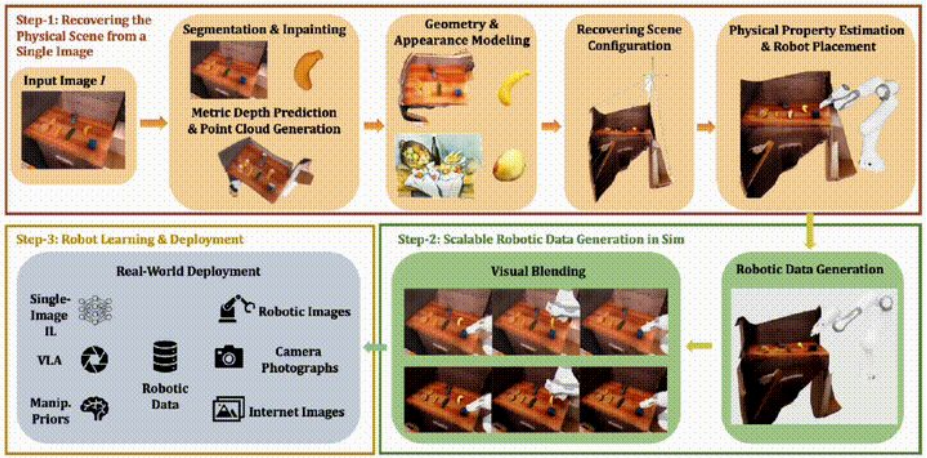
OKAMI: Learning from human videos



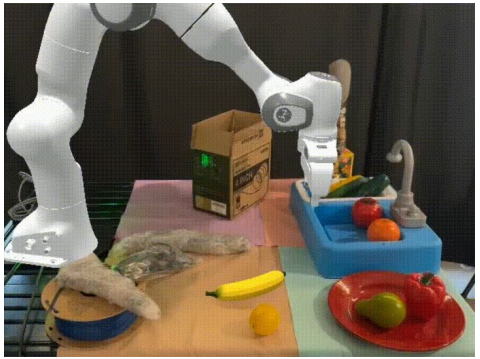
In the second stage, the humanoid motion is synthesized through object-aware retargeting, which retargets the human motion to the humanoid while adapting to the object locations during deployment.

Sim2Real transfer via RoLA

RoLA framework democratizes robotic data generation by producing massive visuomotor robotic demonstrations within minutes from a wide range of image sources.



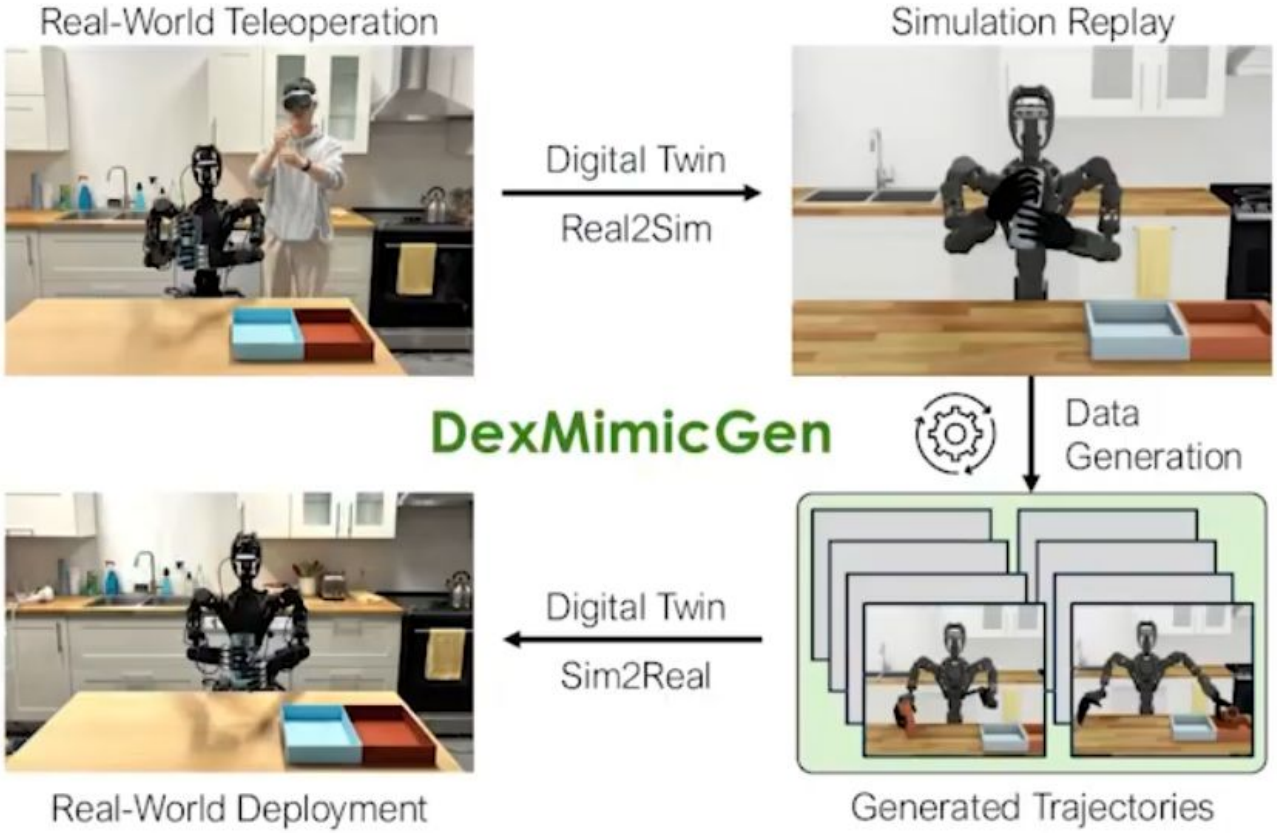
Simulated execution



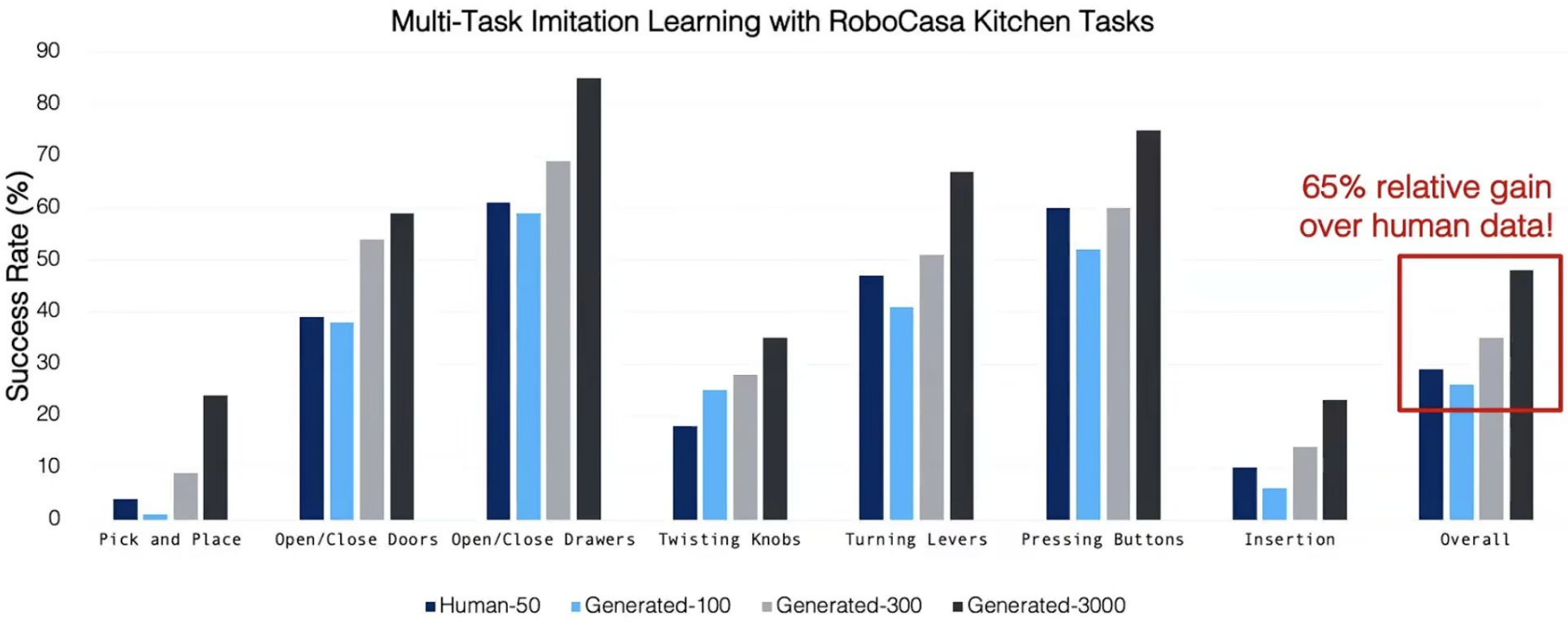
Real-world execution



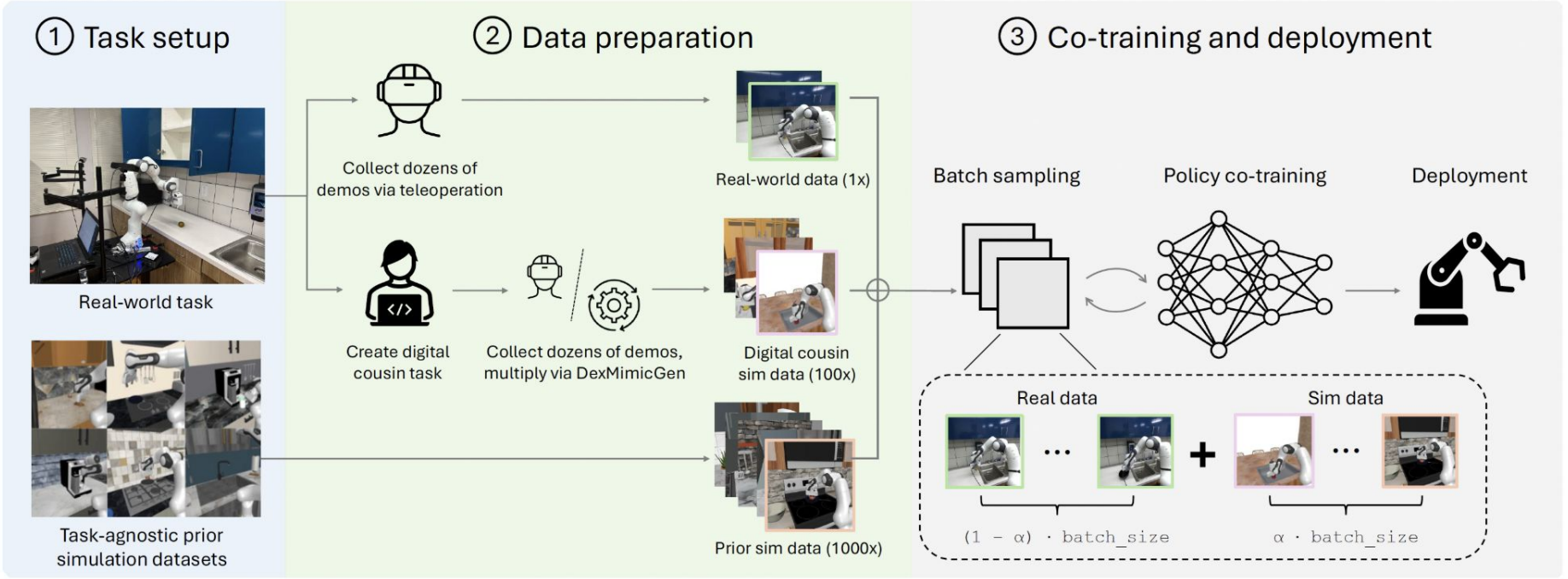
Simulation & Data Collection



DexMimicGen results



Sim-and-Real Co-Training: Learning from Hybrid Data



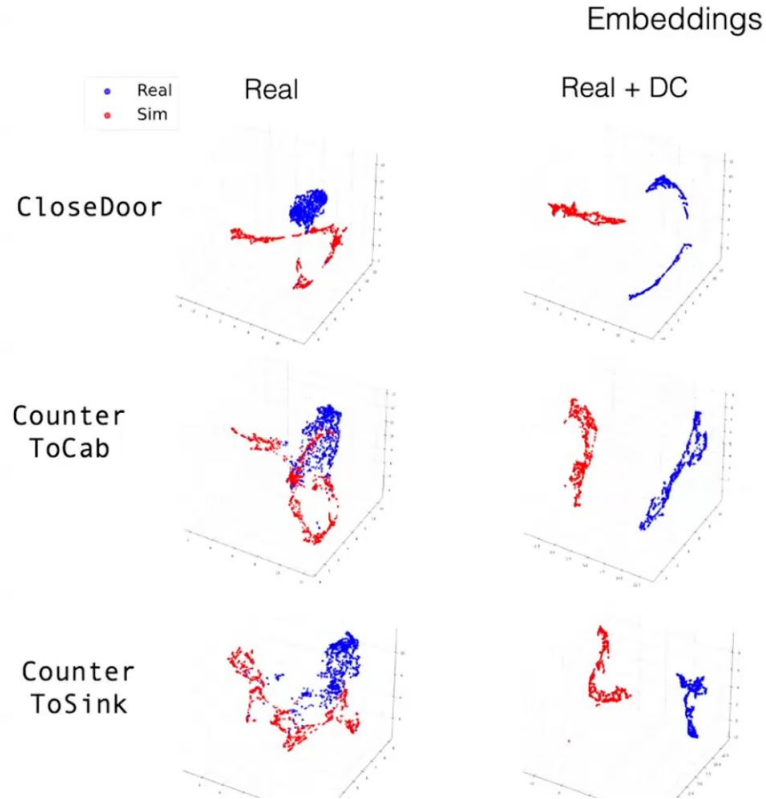
Sim-and-Real Co-Training: Learning from Hybrid Data

The devil is in the details for a successful co-training recipe.

- Enough of both sim and real demonstrations is required.
- Tuning the co-training ratio α is important.
- Camera alignment is critical, but perfect alignment is not required.
- Co-training works in large-scale, multi-task imitation learning settings (more on this later).

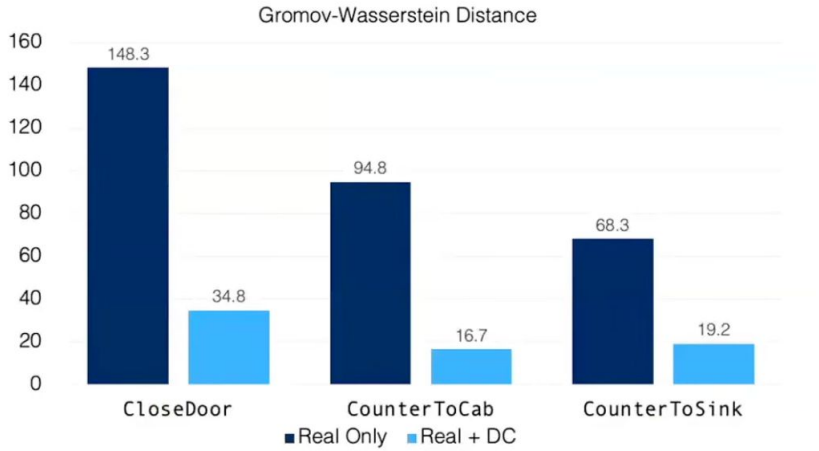


Why Co-Training works? Local alignment in feature space



d_{GW}: Similarity metric for intrinsic geometry of neural features

- $d_{GW} = 0$: represents exactly the same intrinsic geometry
- The smaller d_{GW} is, the more similar geometry two distributions share



Co-Training: Learning across Data Pyramid

Real-Robot Data (1x)

Human Teleoperation



From **Robot Hardware**

- real sensor observations and real motor actions
- costly and time-consuming to collect

Simulated Trajectories (100x)

DexMimicGen



From **Physics Engines**

- rendering gaps and high-quality actions
- physically grounded and annotation-rich

Neural Trajectories (100x)

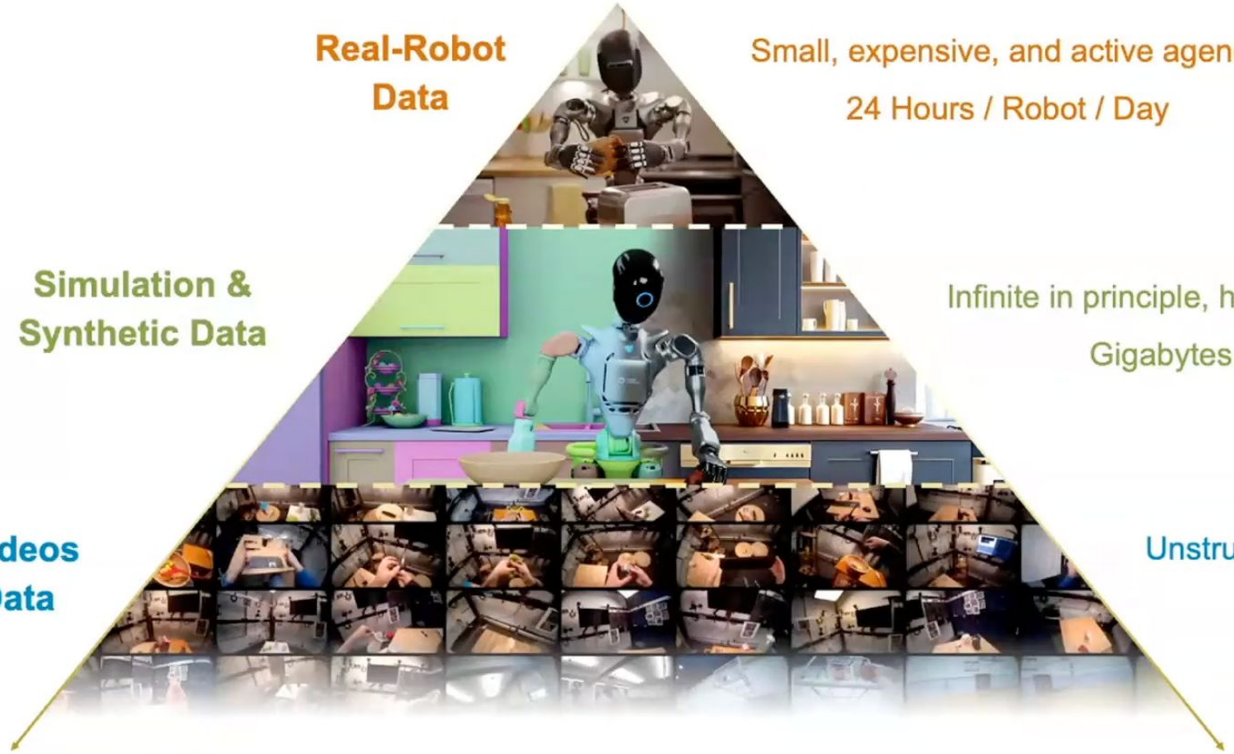
DreamGen



From **Video Generation Models**

- realistic observations and predicted (noisy) actions
- open-ended and simulating complex physics effects

Data Pyramid for Generalist Models



Real-Robot Data

Small, expensive, and active agency

24 Hours / Robot / Day

Simulation & Synthetic Data

Infinite in principle, hard to scale in practice

Gigabytes / GPU / Day

Human Videos & Web Data

Unstructured, multimodal, passive

Exabytes / Day