

# AI for XR

## Crafting Intelligent XR Applications with AI

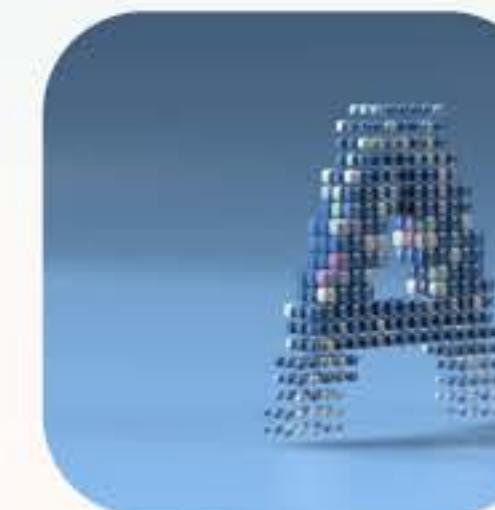
Tamil Selvan Gunasekaran

# Session Agenda

---



Understand how AI transforms XR experiences.

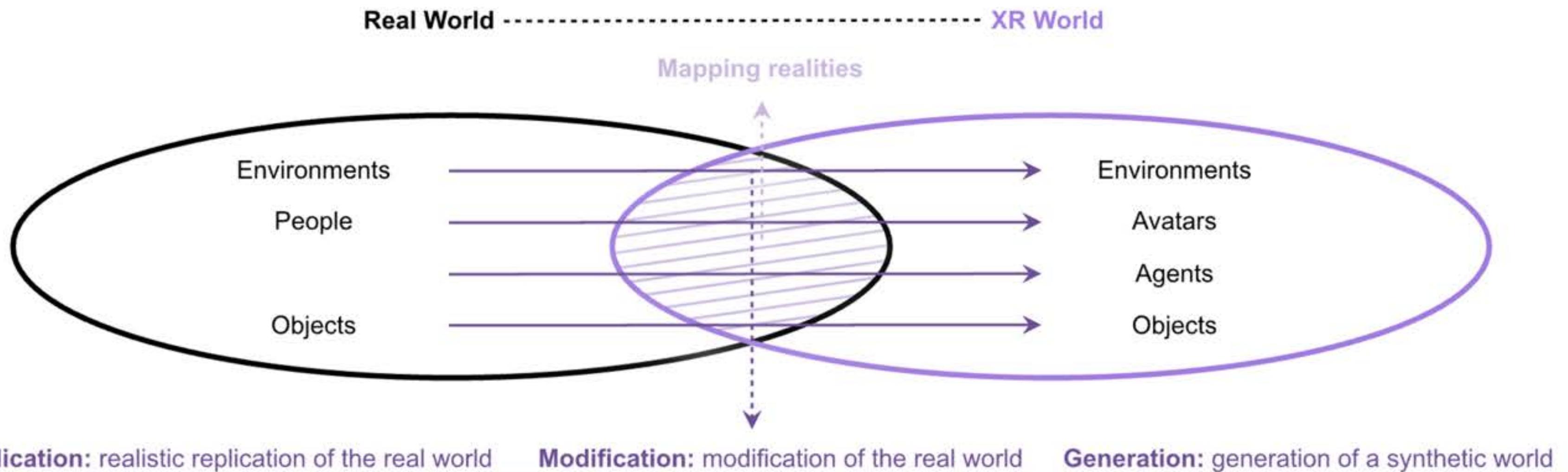


Build practical AI-driven XR interactions using Unity.



Explore cutting-edge XR+AI tools and APIs."

# How is AI used to build XR ?

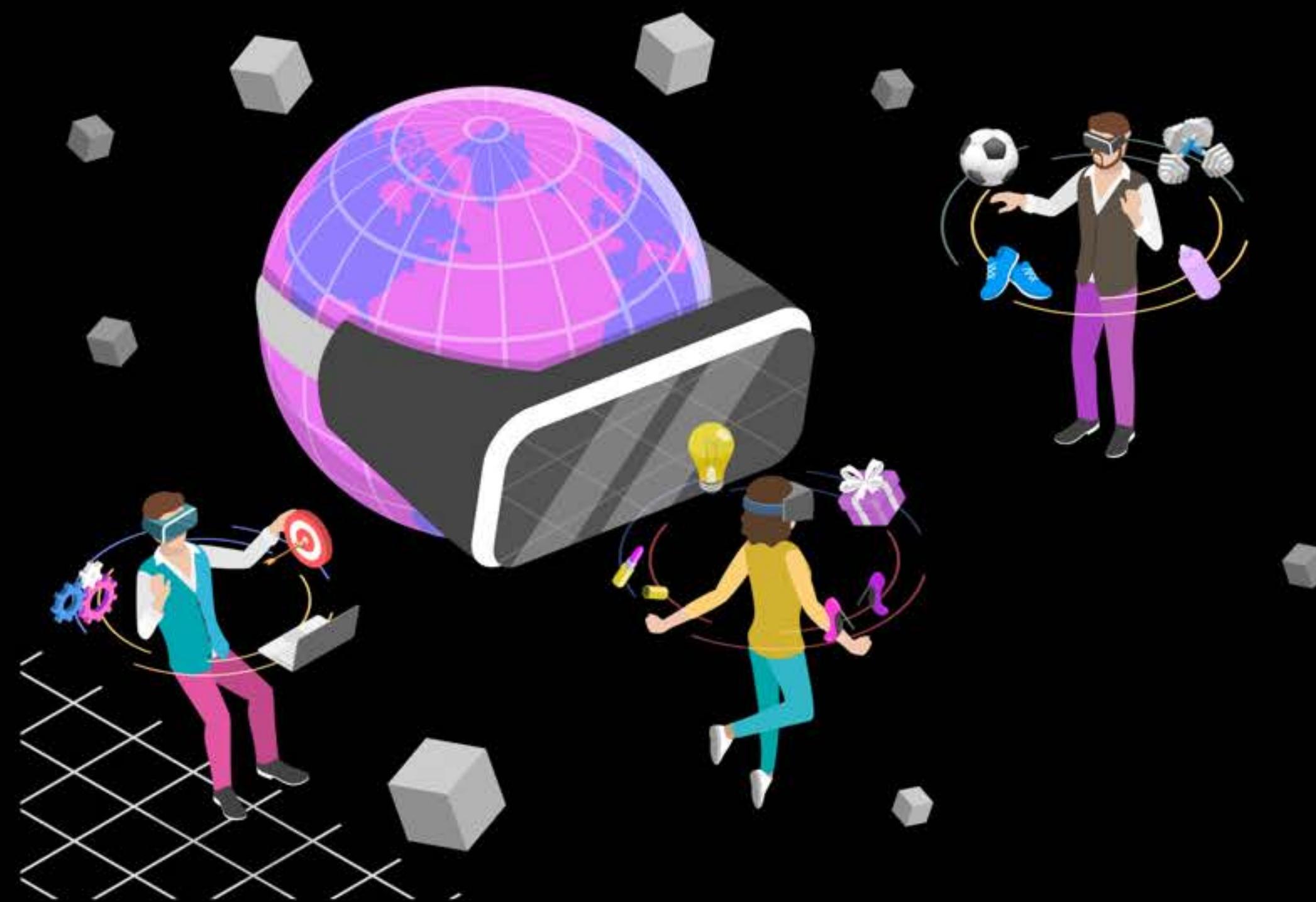


Hirzle, T., Müller, F., Draxler, F., Schmitz, M., Knierim, P., & Hornbæk, K. (2023, April). When xr and ai meet-a scoping review on extended reality and artificial intelligence. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (pp. 1-45).

# **State-Of-The-Art XR and AI Research**

- (1) Using AI to create XR worlds
- (2) Using AI to understand users in XR
- (3) Using AI to support interaction in XR
- (4) Interaction with Intelligent Virtual agents
- (5) Using XR to support AI research

# AI to create XR worlds



# Using AI to Create XR Worlds

---

1

Creating XR  
Environments

2

Creating  
Avatars

3

Creating  
Agents

4

Creating XR  
Objects

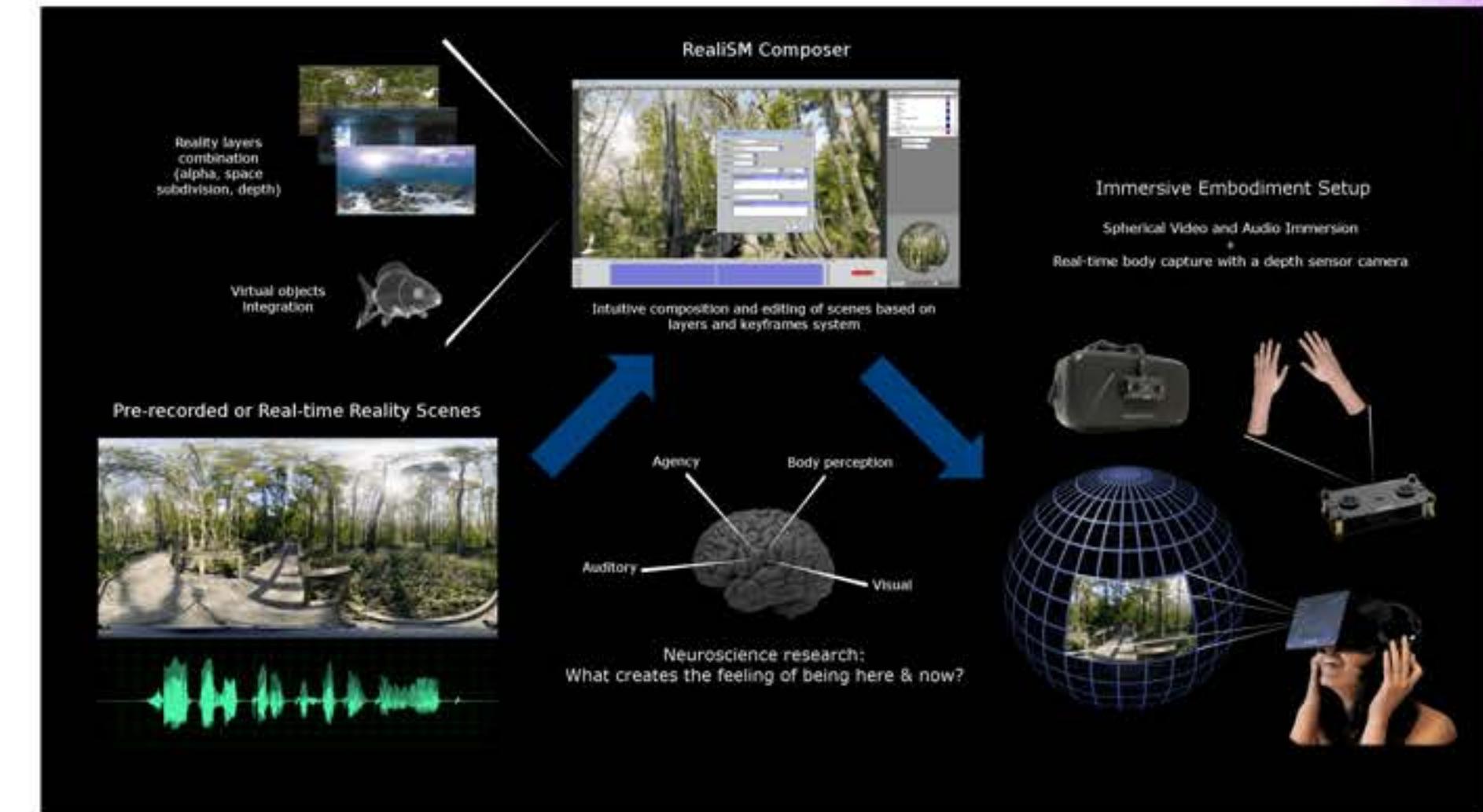
# Creating XR Environments with AI

## 1. Tracking of Environments: AI methods for real-time environment tracking.

- Example: SLAM (Simultaneous Localization and Mapping).

## 2. Presenting Realistic Virtual Content: Techniques for realistic rendering and spatial audio.

- Example: GANs for high-quality textures.



# Creating XR Environments with AI

**3. Measuring and Optimizing Illumination:** AI models for dynamic lighting and shadow adjustments.

- Example: CNNs for realistic lighting conditions.



**4. Optimizing Image Quality:** AI-based super-resolution techniques.

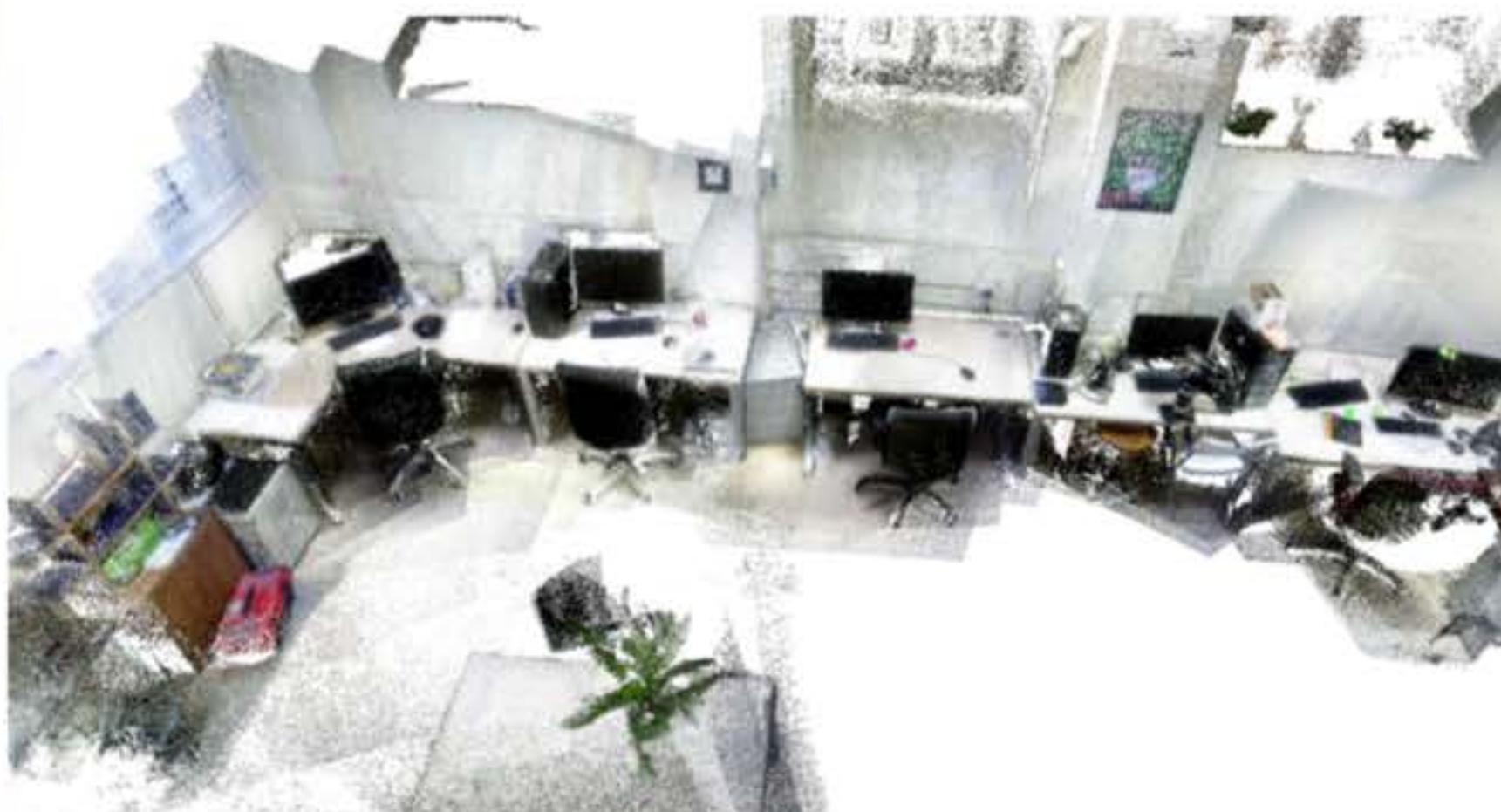
- Example: Deep learning for enhancing image details.



# Creating XR Environments with AI

## 5. Mapping and Augmenting Environments: Using AI for environment mapping and AR integration.

- Example:
  - Generative 3D exploded: NeRF variants now train & render in seconds and 3D Gaussian Splatting brings 100 + fps real-time view-synthesis.
  - Meta's Segment-Anything (SAM) provides one-click object masks for AR/VR pipelines



# Creating Avatars

- 1. Recognition and Animation of Facial Expressions:** AI models for capturing and replicating facial expressions.
  - Example: Deep learning for real-time facial animation.
- 2. Physical Appearance: Aspects of Human Bodies:** Modeling realistic body structures and movements.
  - Example: AI for lifelike body gestures.

Motivation  
Visual Quality Metrics



HumanGPS  
Process



12061 ms

# Creating Avatars

## 3. Tracking and Full Body Reconstruction:

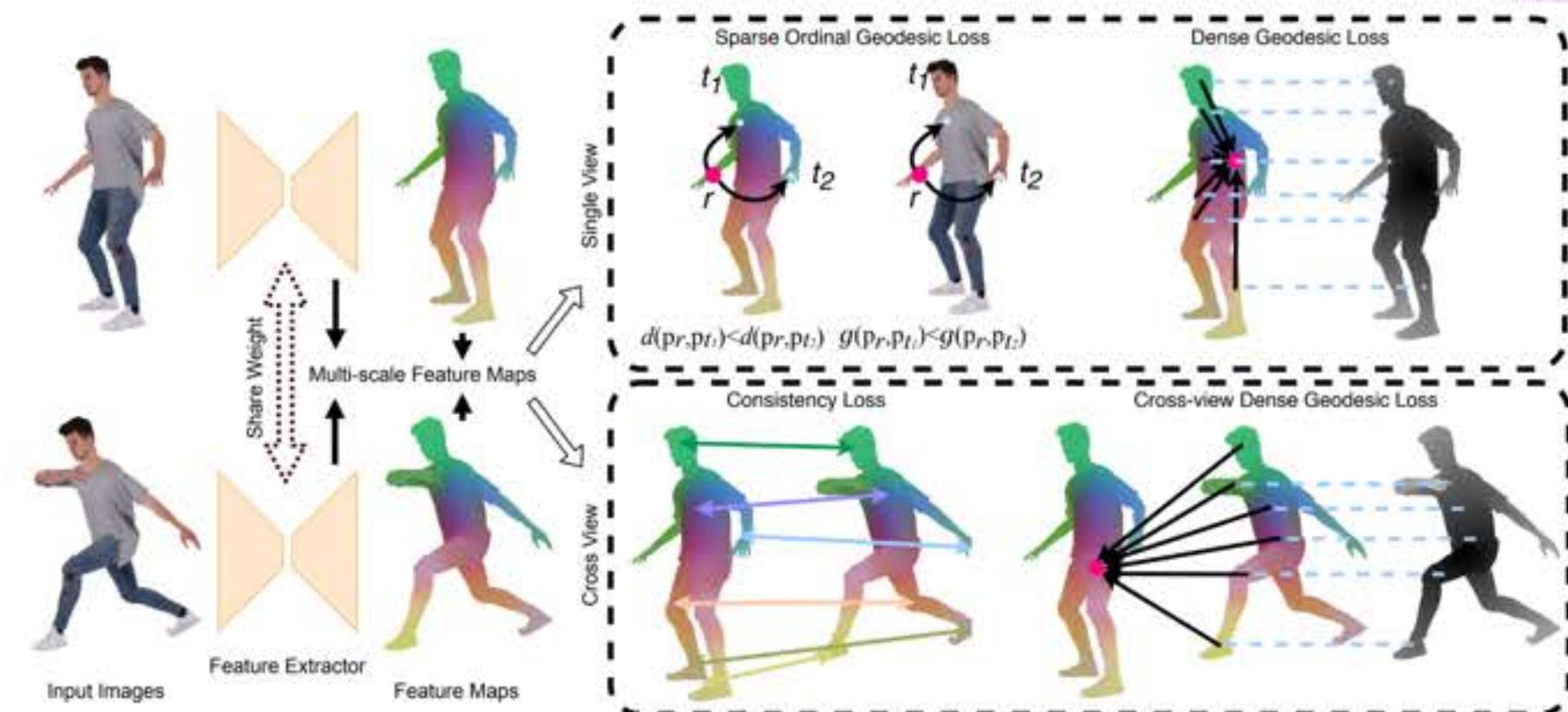
**AI-driven body tracking and 3D reconstruction.**

- Example: Motion capture for creating avatars.

## 4. Animation and Movement Modeling:

**AI for realistic animation of movements and gestures.**

- Example: Neural networks for gesture recognition.



# Creating Agents

## 1. Realistic Modeling of Agent Behaviour:

AI techniques for creating believable agent behaviors.

- Example: Reinforcement learning for adaptive behaviors



## 2. Realistic Modeling of Agent Behavior:

AI techniques for creating believable agent behaviors.

- Example: LLM-driven “generative agents” simulate believable NPCs that plan, remember, converse and co-create stories in sandbox towns

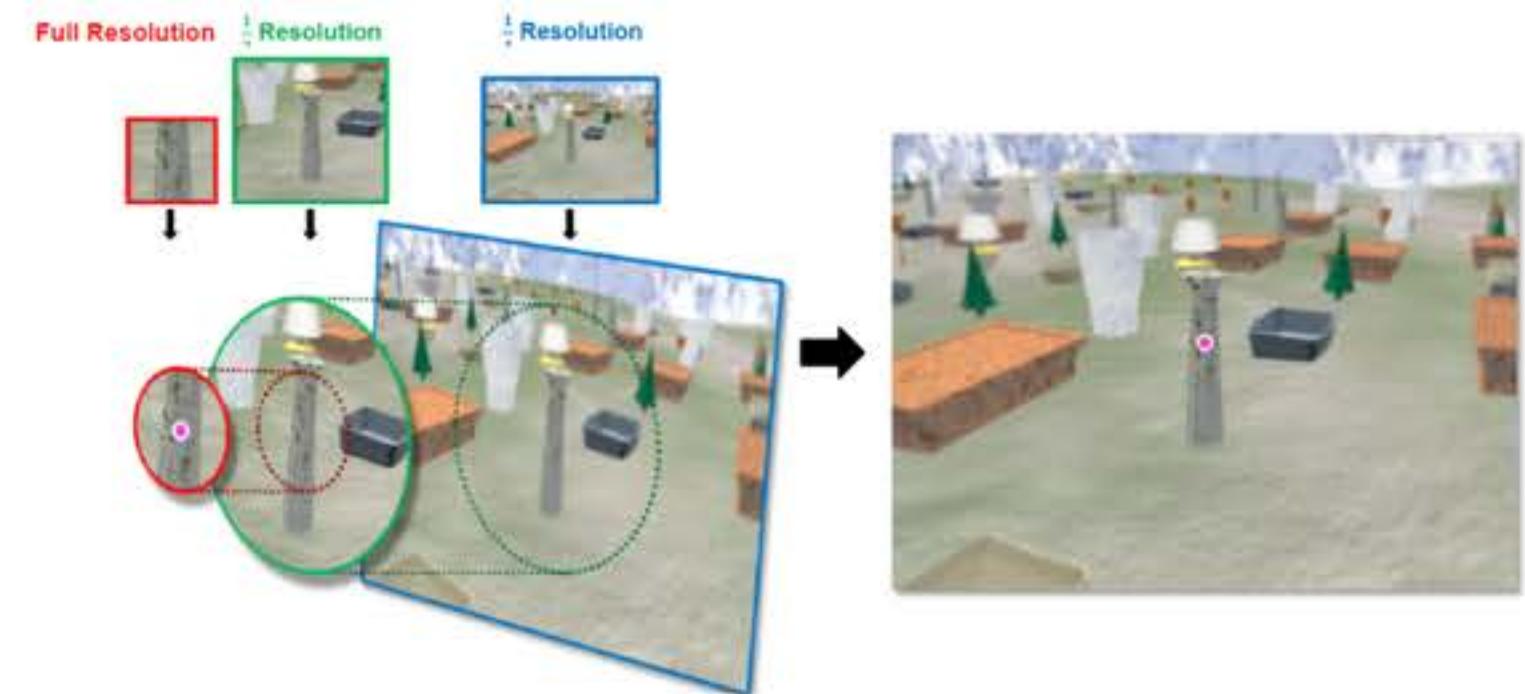
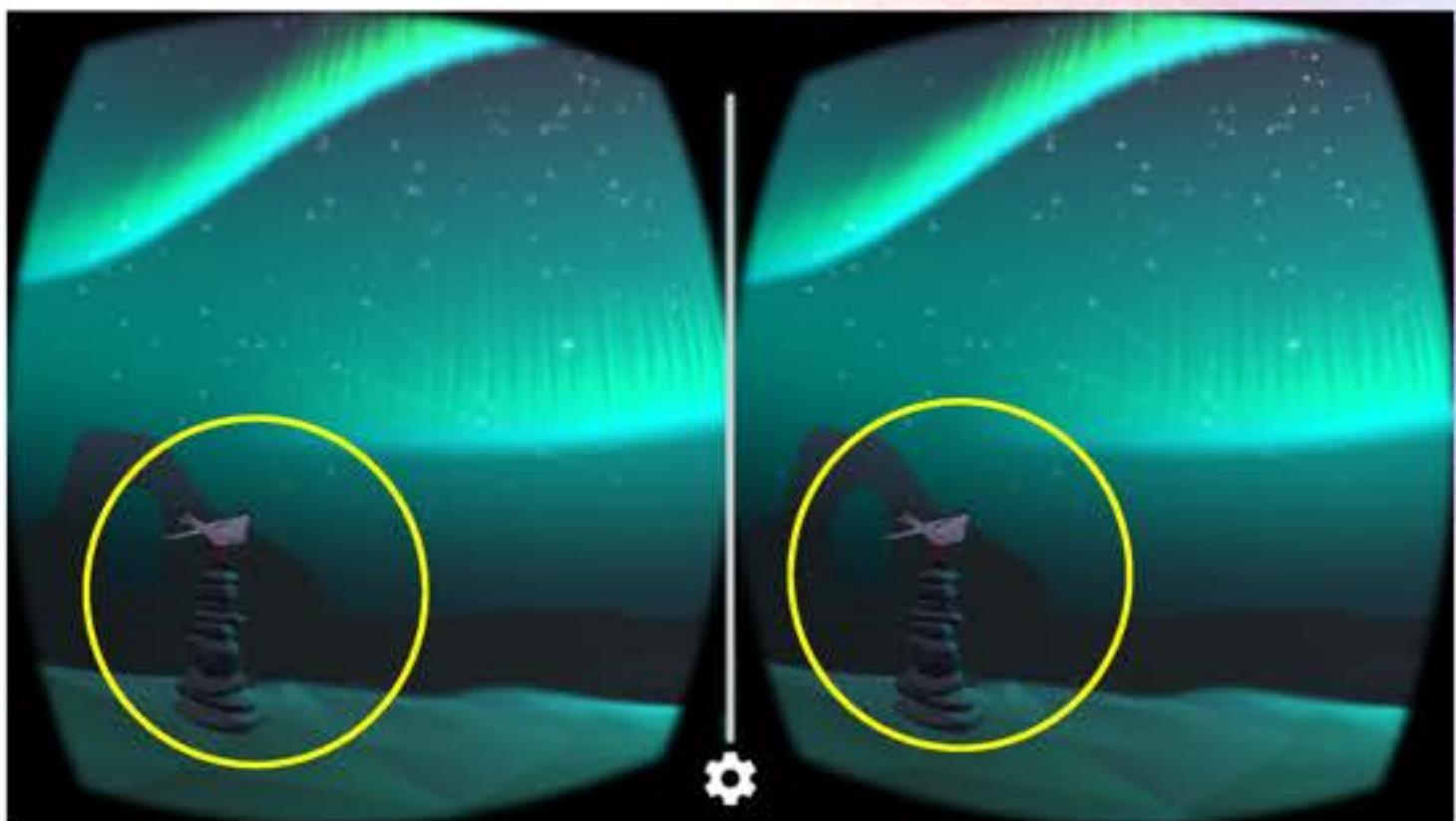
# Creating XR Objects

**1. Tracking of Objects:** AI models for real-time object tracking in XR.

- Example: Object detection using deep learning.

**2. Rendering and Modifying Objects:** AI for realistic rendering and modification of virtual objects.

- Example: Texture transfer and object appearance adaptation.



# AI to understand users in XR



# Enhancing User Interactions in XR with AI

AI can significantly improve user interactions by making them more intuitive, responsive, and adaptive to the user's needs.

- Foundation multimodal models (e.g., Gemini, GPT-4o) now run on-device, summarising screen context, voice & gaze in real time



# Using AI to Understand Users

---



Predicting VR Sickness



Predicting User  
Characteristics:



Eye Tracking and Gaze  
Analysis

# Predicting VR Sickness

- 1. Predictive Models:** AI models for predicting user discomfort and VR sickness.
  - Example: LSTM networks for real-time prediction.
- 2. Adaptive Techniques:** AI-driven methods for reducing VR sickness by adjusting VR parameters.
  - Example: Dynamic field-of-view adjustments.



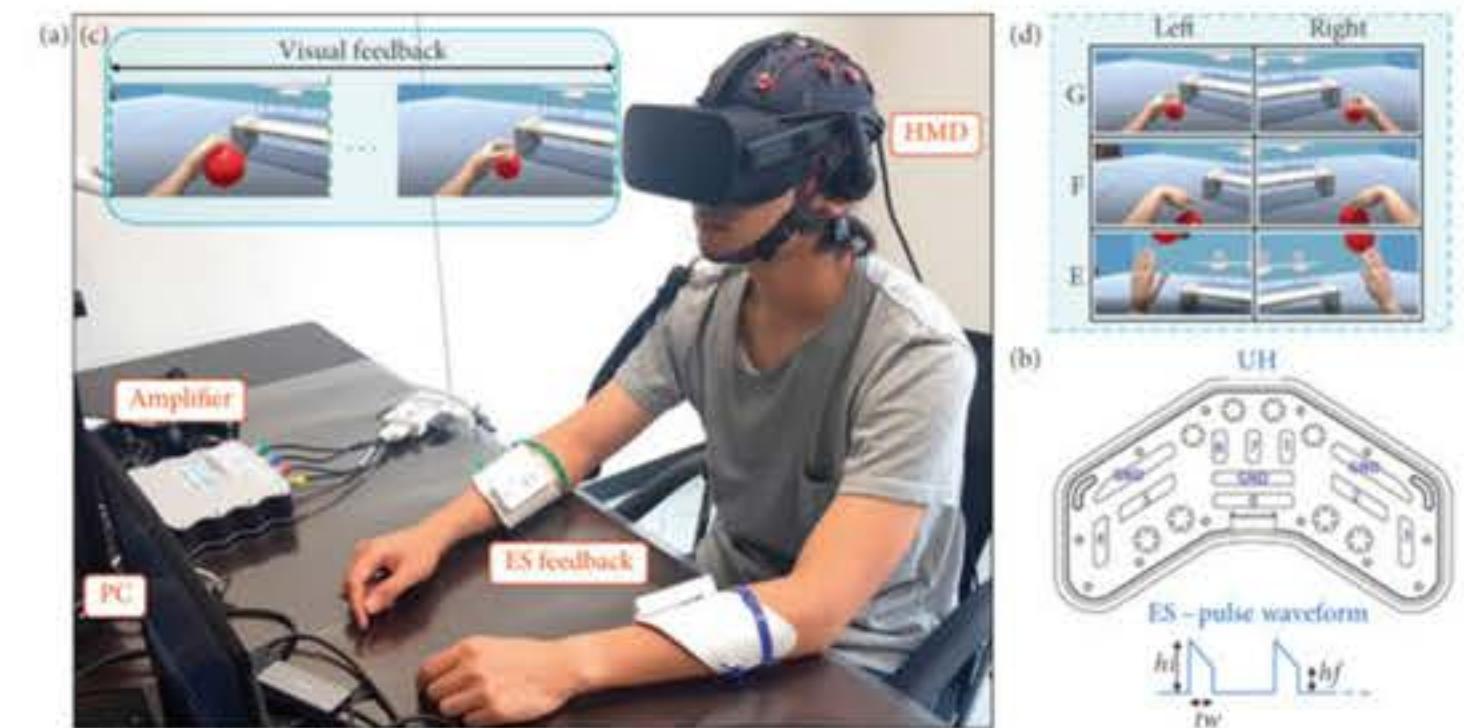
# Predicting User Characteristics

**1. User Profiling:** AI techniques for real-time user profiling based on behavior and interactions.

- Example: Clustering and classification methods for user segmentation.

**2. Predicting Viewport and Head Movement:** AI models for predicting user head movements and viewport changes.

- Example: Reinforcement learning for optimal viewport prediction.



# Eye Tracking and Gaze Analysis

## 1. Gaze Analysis and Visual Attention

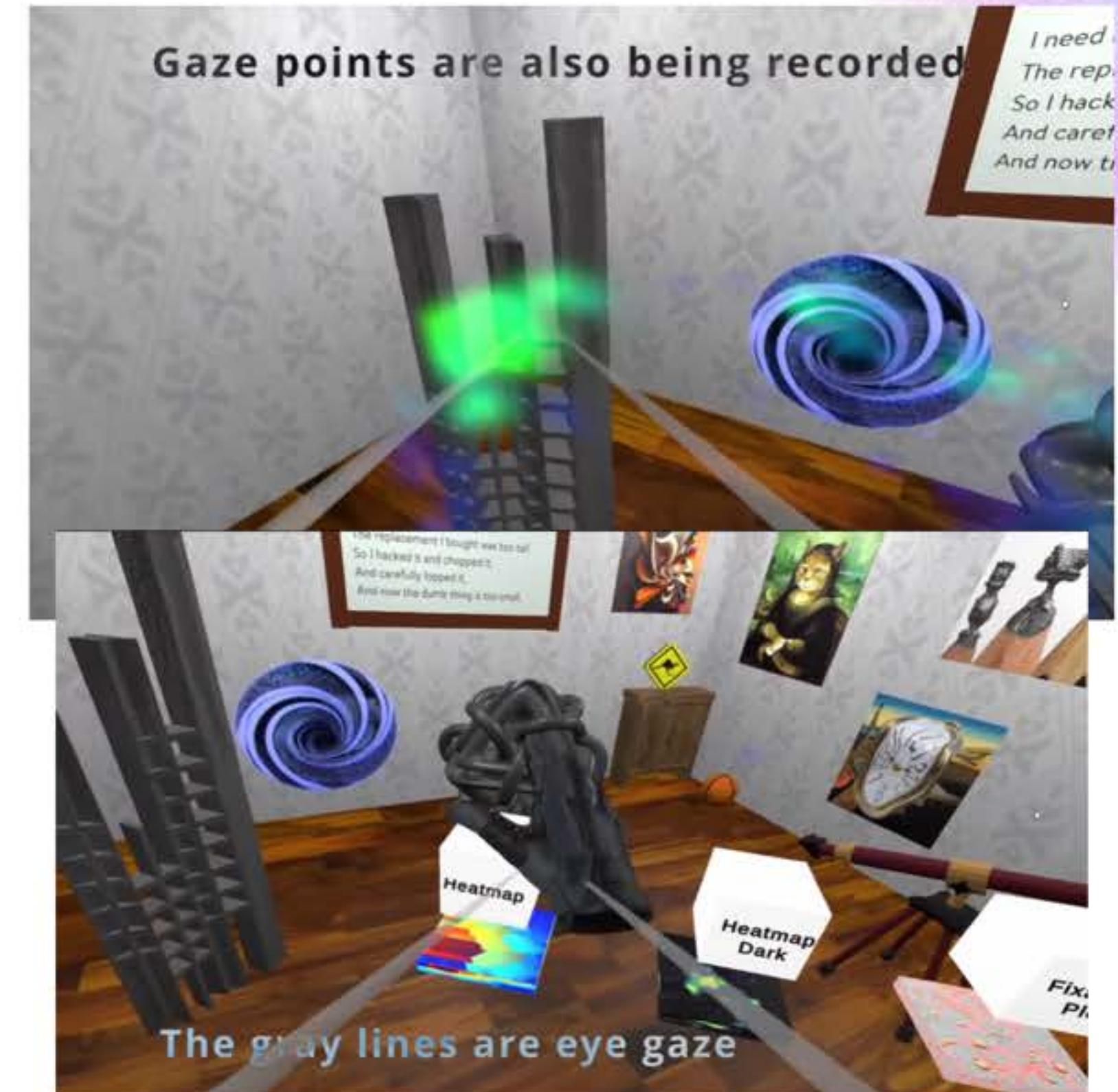
**Estimation:** AI models for tracking gaze and estimating visual attention in XR.

- Example: CNNs for gaze point estimation.

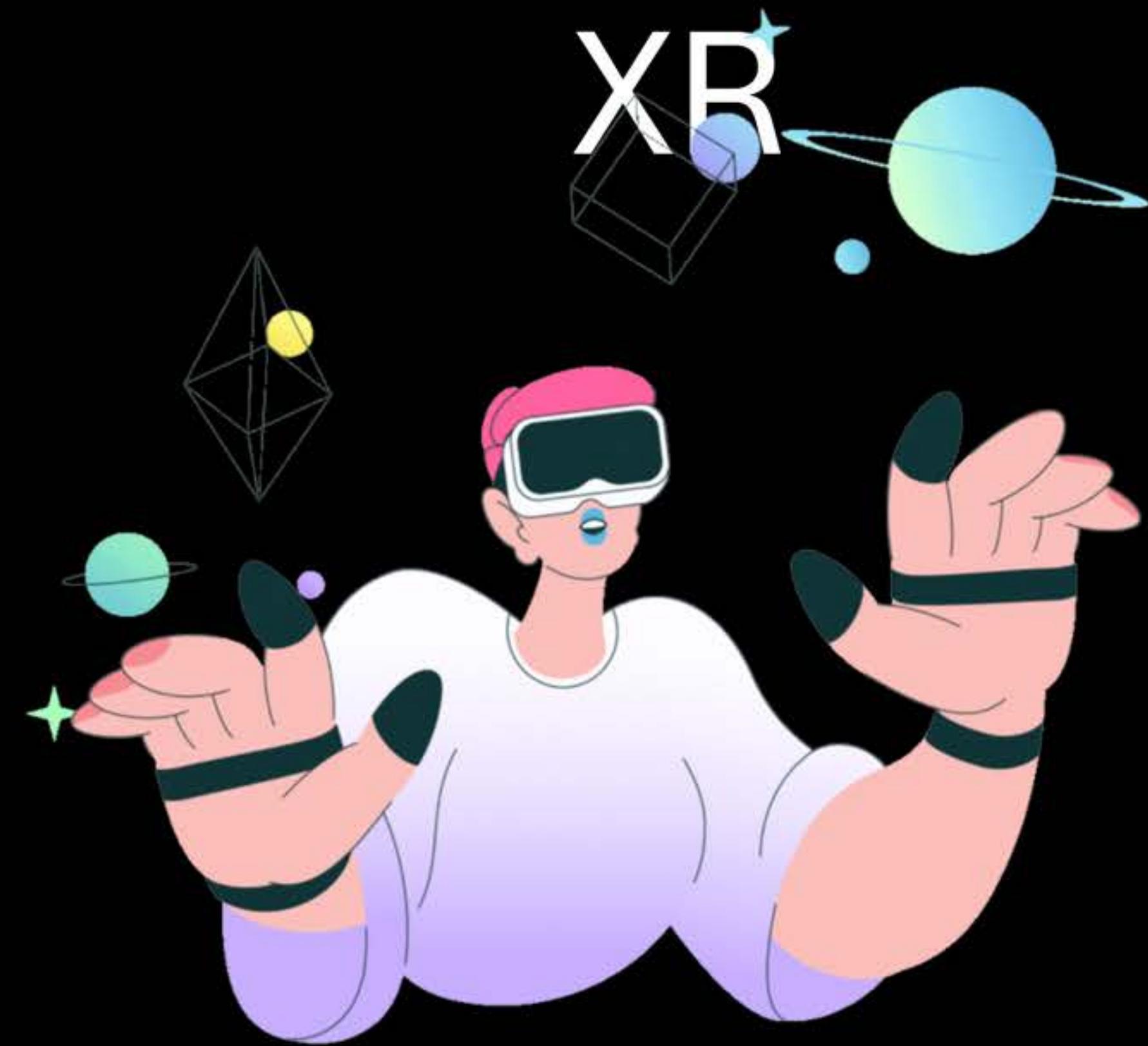
## 2. Gaze Prediction and

**Modelling:** Predictive modeling of gaze patterns based on user behavior.

- Example: LSTM networks for real-time gaze prediction.



# AI to support interaction in



# Using AI to Support Interaction

---

1

Gesture-based  
Interaction

2

Locomotion  
Techniques

3

Novel Devices  
and Interaction  
Techniques

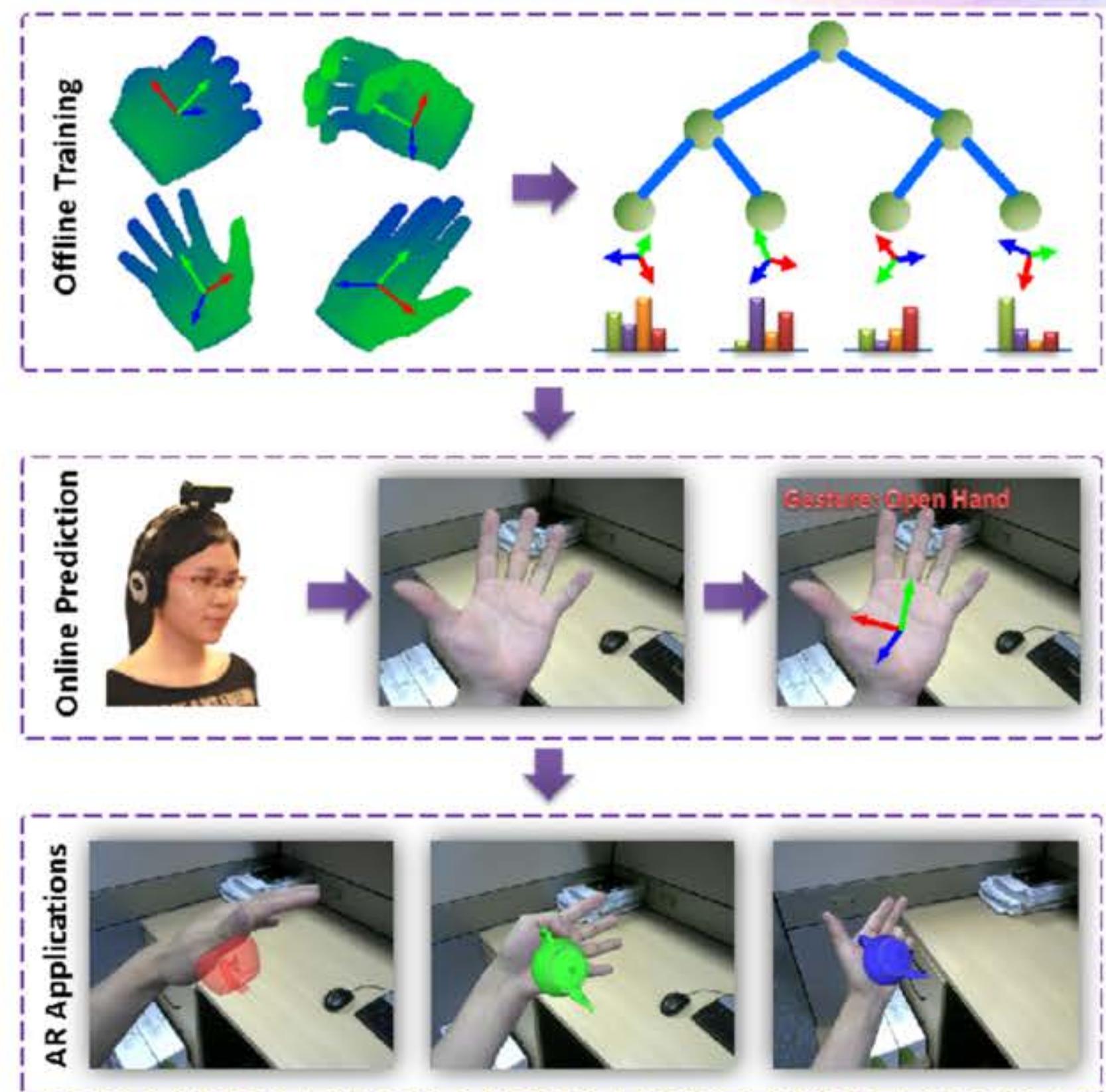
4

Haptic  
Feedback

# Gesture-Based Interaction

1. **3D Mid-Air Gesture Interaction:** AI methods for recognizing and responding to mid-air gestures.
  - Example: CNNs for hand gesture recognition.

2. **Gesture Recognition and Classification:** AI models for classifying complex gestures in real-time.
  - Example: LSTM networks for dynamic gesture classification.



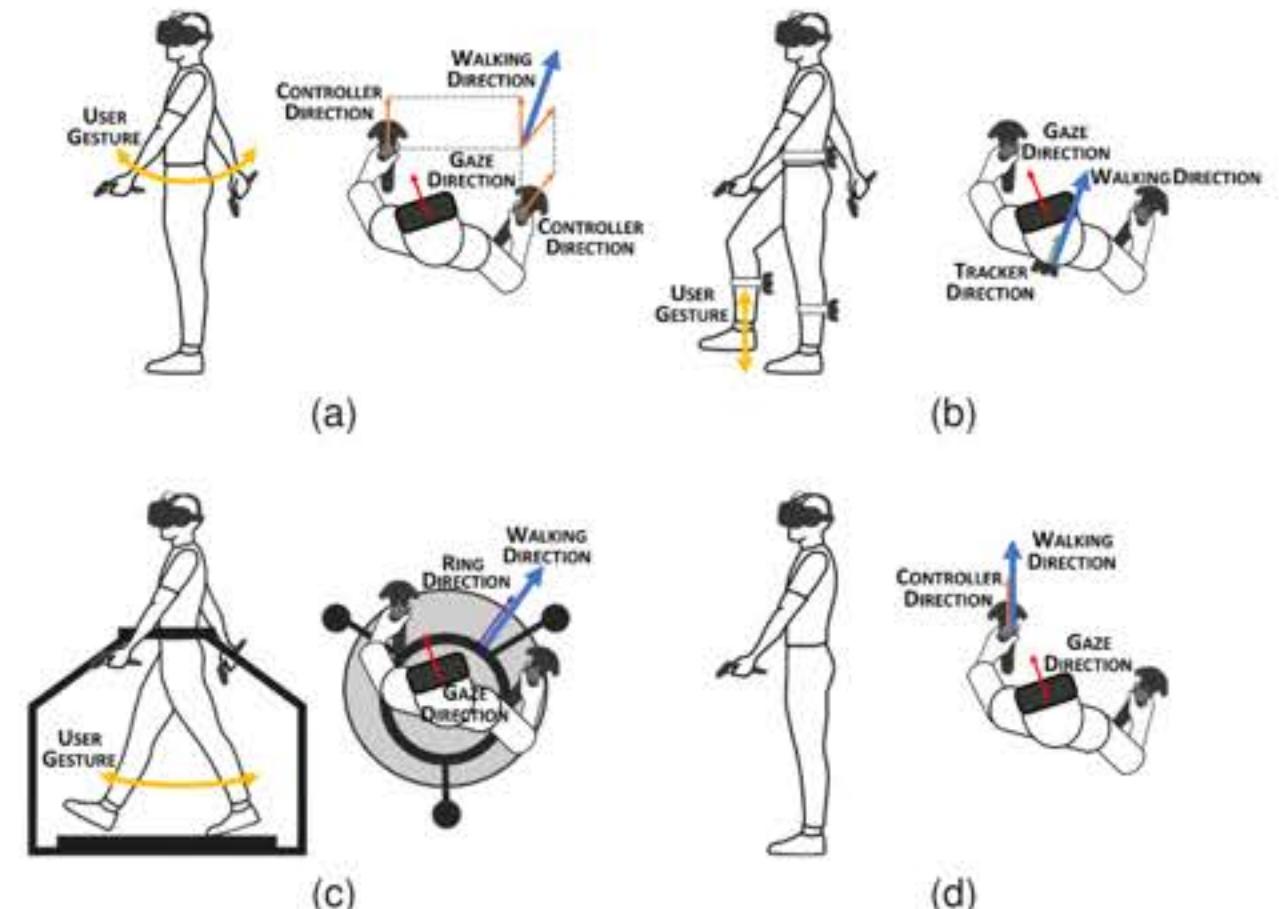
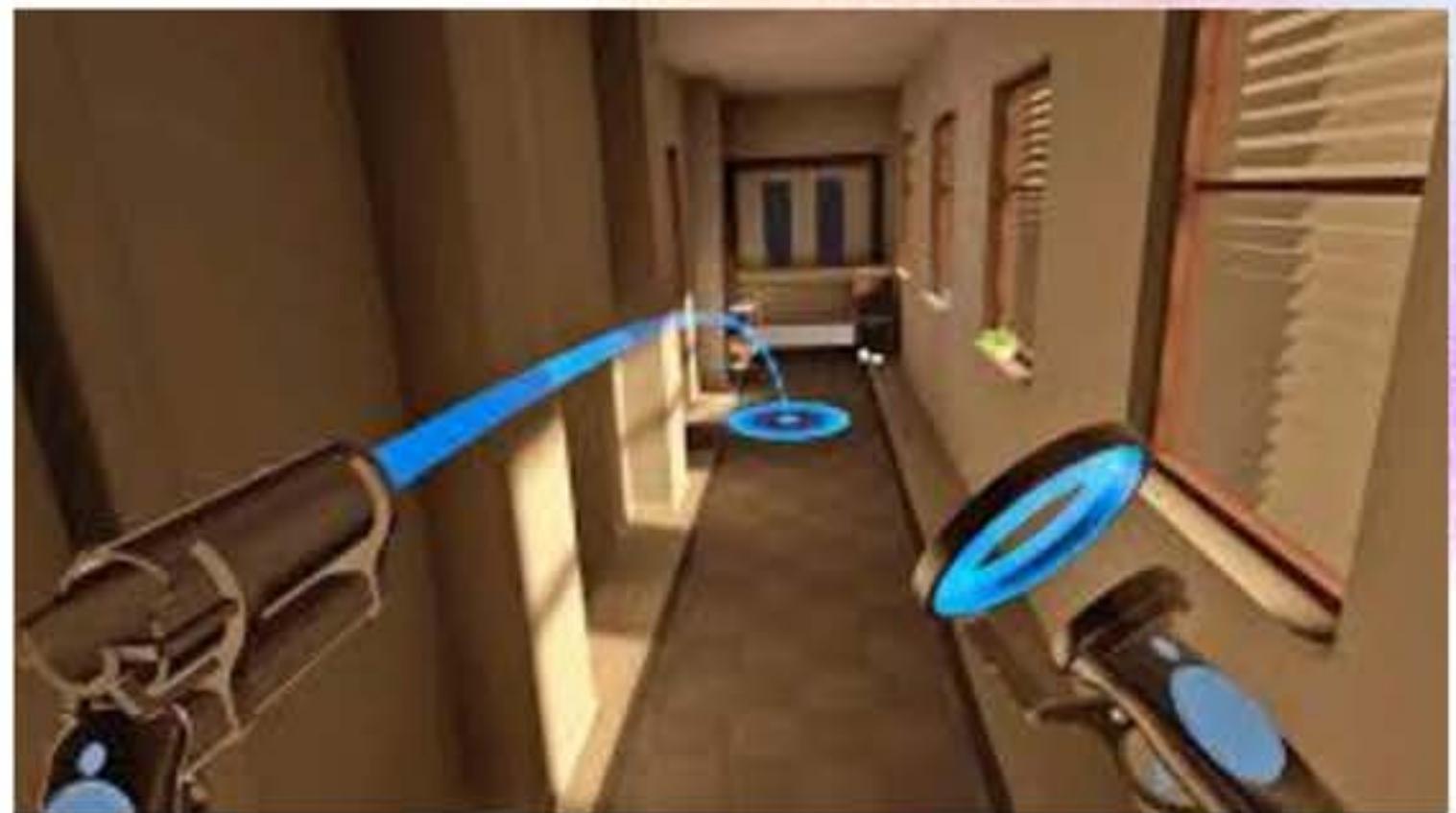
# Locomotion Techniques

**1. Redirected Walking Techniques:** AI models for seamless navigation in confined VR spaces.

- Example: Reinforcement learning for optimizing redirected walking paths.

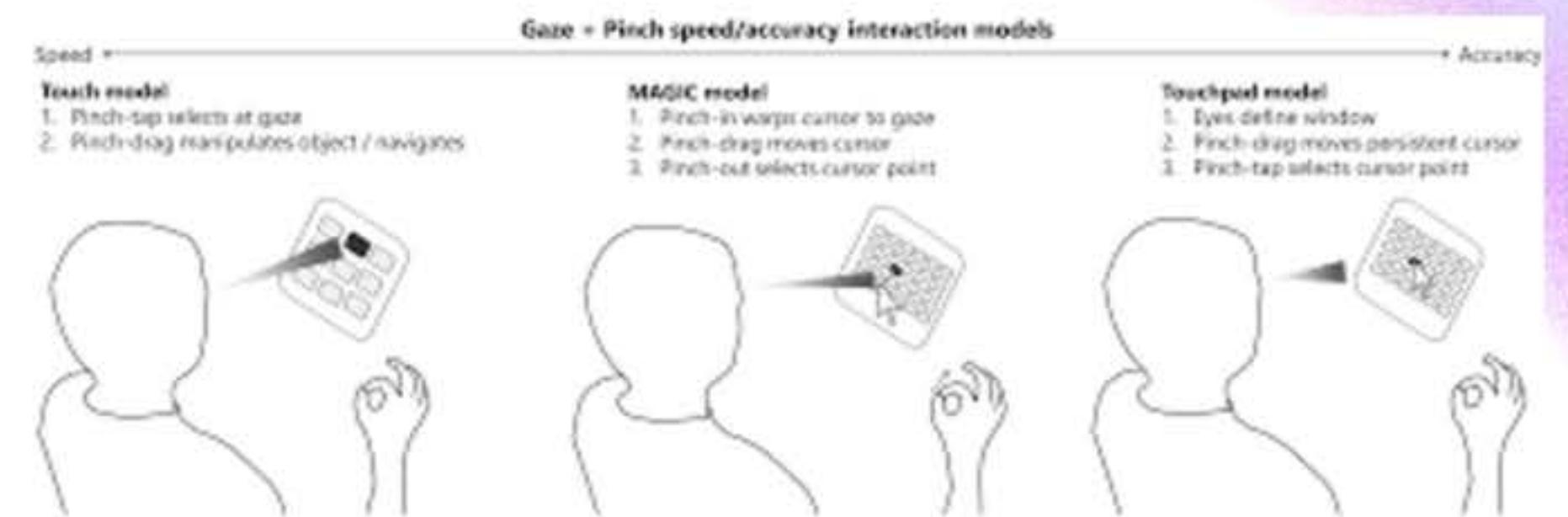
**2. General Locomotion Techniques:** AI-enhanced techniques for natural and intuitive locomotion.

- Example: Predictive models for dynamic path adjustment.



# Novel Devices and Interaction Techniques

- **Novel Devices:** AI-enhanced devices for novel interaction modalities in XR.
- Example: Smart controllers and adaptive interfaces.

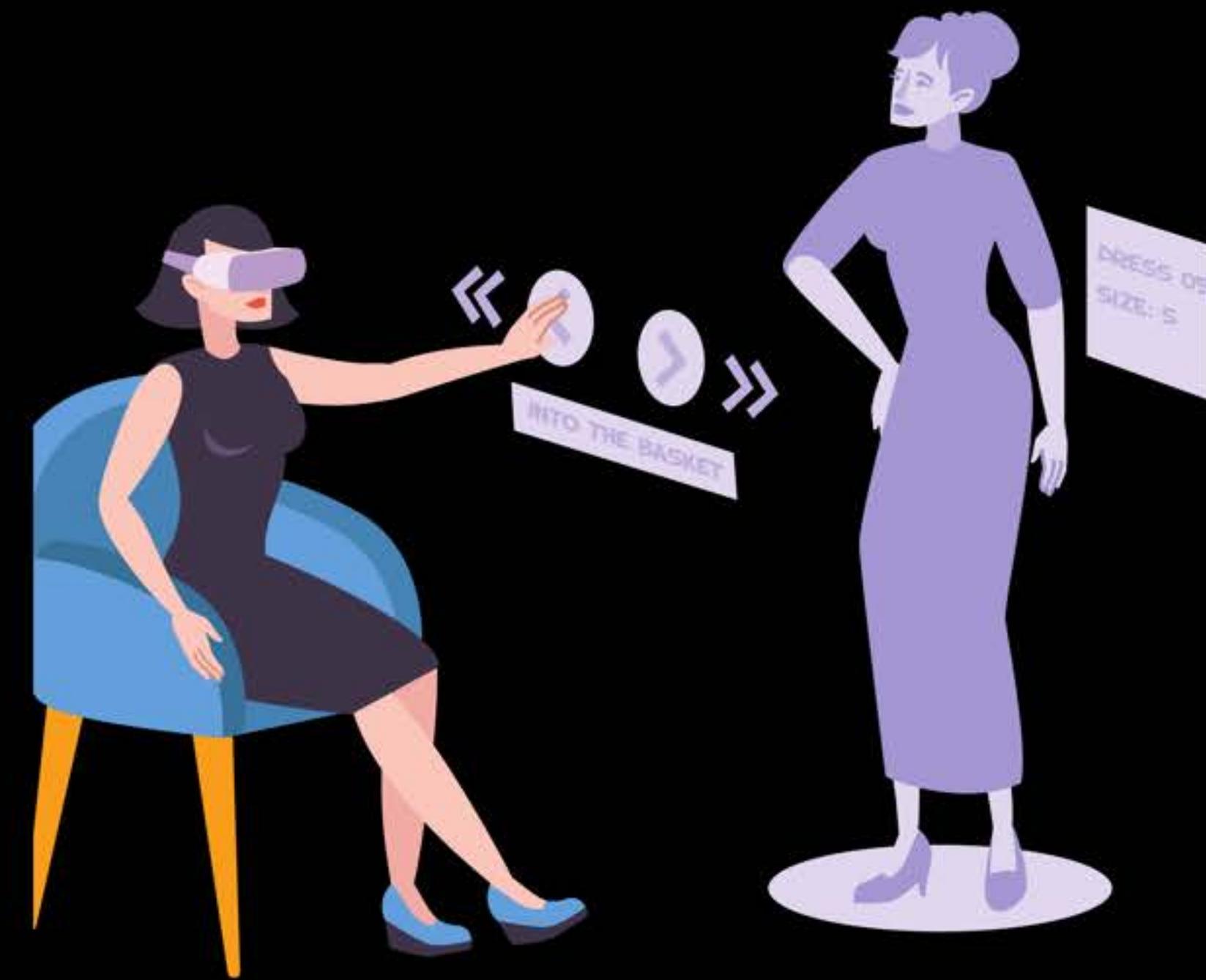


# Haptic Feedback

- AI models for enhancing haptic feedback based on user interactions.
  - Example: AI-driven tactile feedback for realistic object manipulation.

System Overview	HaptX DK2	HaptX G1	Meta Haptic Glove	Our System
Glove Closeup				
Haptic Arrays				
	<b>Weight:</b> >30kg <b>Cost:</b> ~\$75K <b>Pixels:</b> 133/glove <b>Power:</b> Wall <b>Glove:</b> Tethered (B&C)	<b>Weight:</b> 17kg <b>Cost:</b> \$6K+\$500/mo <b>Pixels:</b> 135/glove <b>Power:</b> Backpack <b>Glove:</b> Tethered (B)	<b>Weight:</b> >25kg <b>Cost:</b> >\$15K <b>Pixels:</b> 15/glove <b>Power:</b> Wall <b>Glove:</b> Tethered (B&C)	<b>Weight:</b> 0.2kg <b>Cost:</b> <\$1K <b>Pixels:</b> 160/glove <b>Power:</b> Glove Battery <b>Glove:</b> Untethered

# Interaction with Intelligent Virtual agents



# Interaction with Intelligent Virtual Agents



Interacting with Agents



Physical Interaction with Agents



Trust and Empathy in Agents

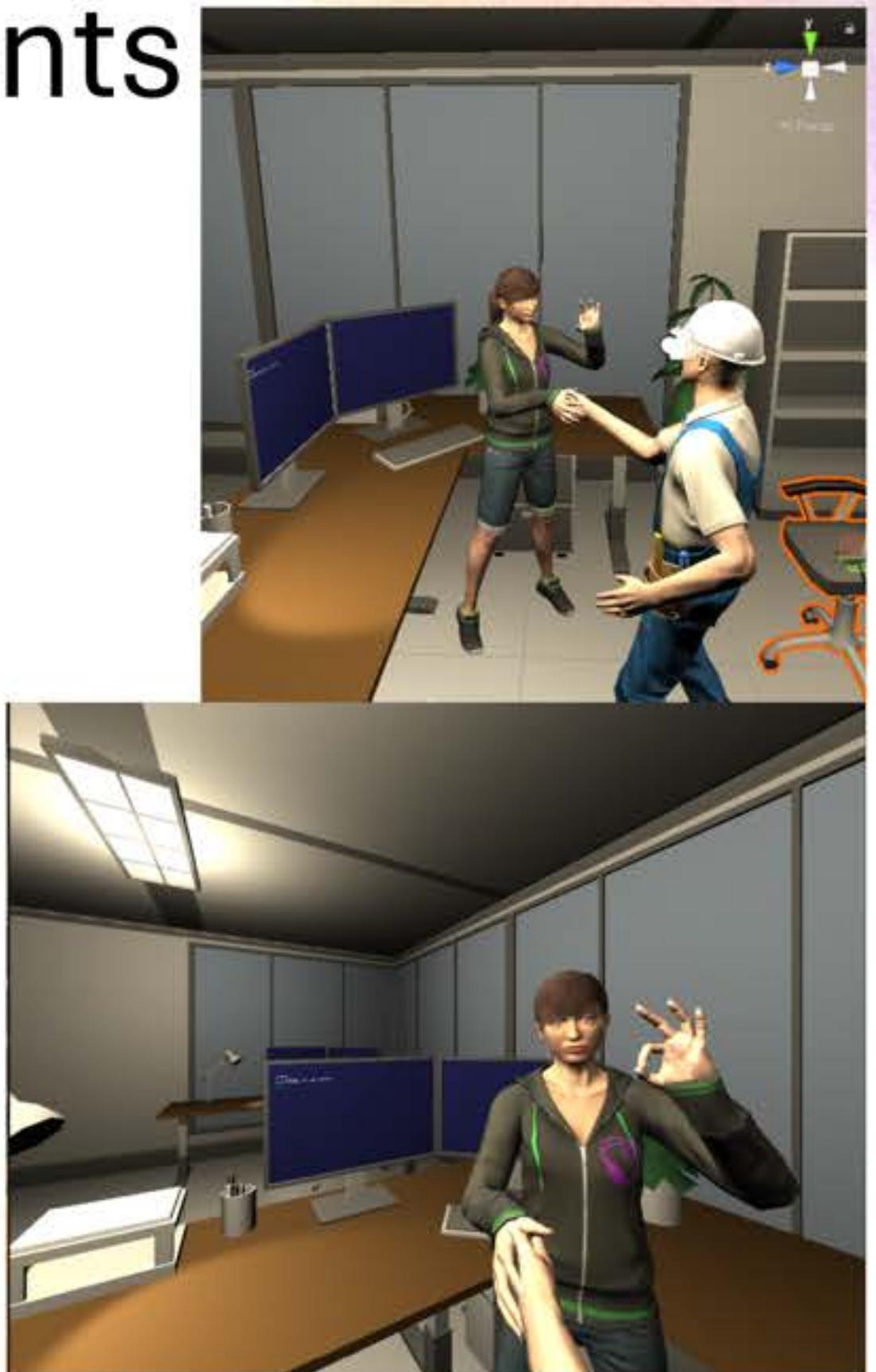
# Interacting with Crowds of Agents

- **Crowd Behavior Modeling:** AI models for realistic crowd behaviors and dynamics.
  - Example: Multi-agent reinforcement learning for crowd simulation.



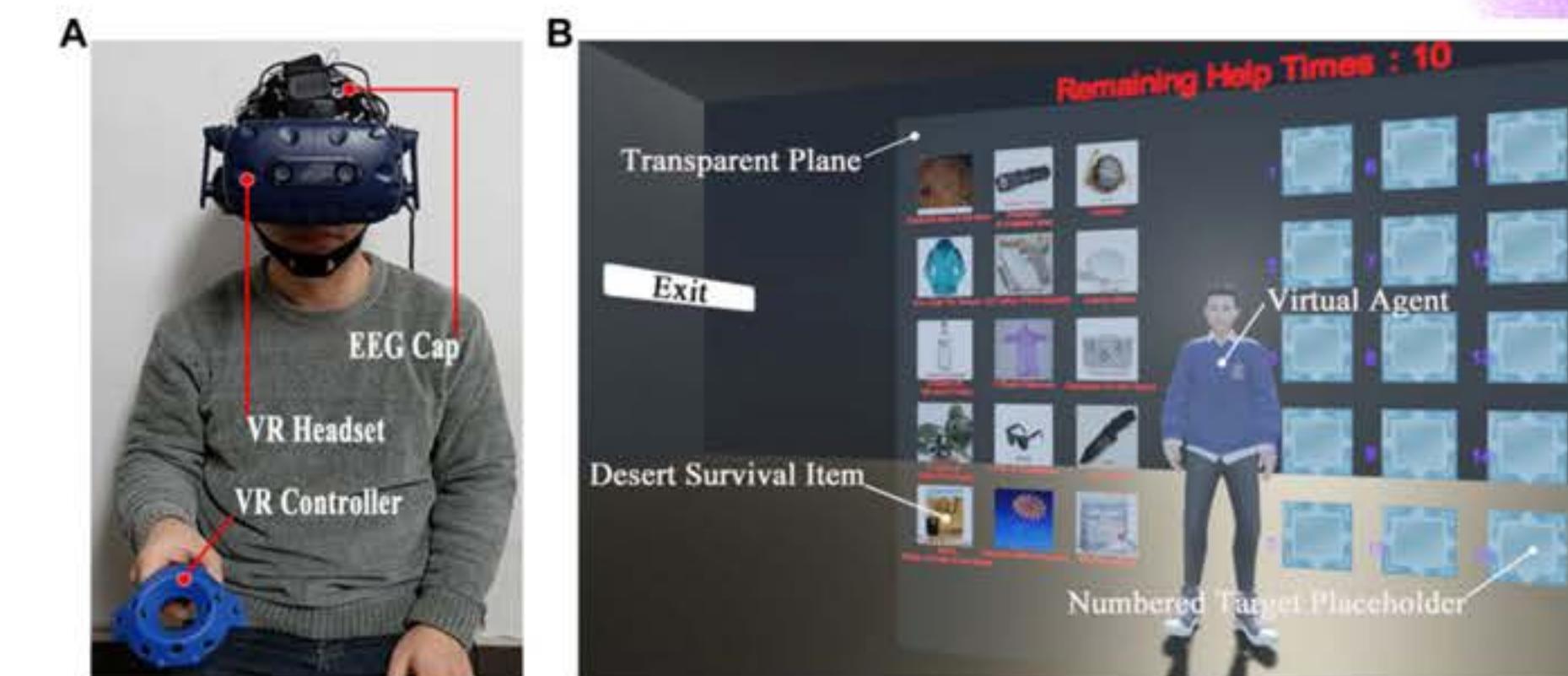
# Physical Interaction with Agents

- **Peripersonal Space and Touch:** AI models for realistic physical interactions and touch perception.
  - Example: AI for simulating haptic feedback during interaction.
- **Touch-Based Interaction:** AI for touch-based interactions such as handshakes and object exchanges.
  - Example: Haptic feedback for realistic touch sensations.



# Trust and Empathy in Agents

- **Trust in Agents:** AI methods for enhancing trust through realistic and empathetic behaviors.
- Example: Adaptive behaviors based on user trust levels.



# XR to support AI research



# Using XR to Support AI Research



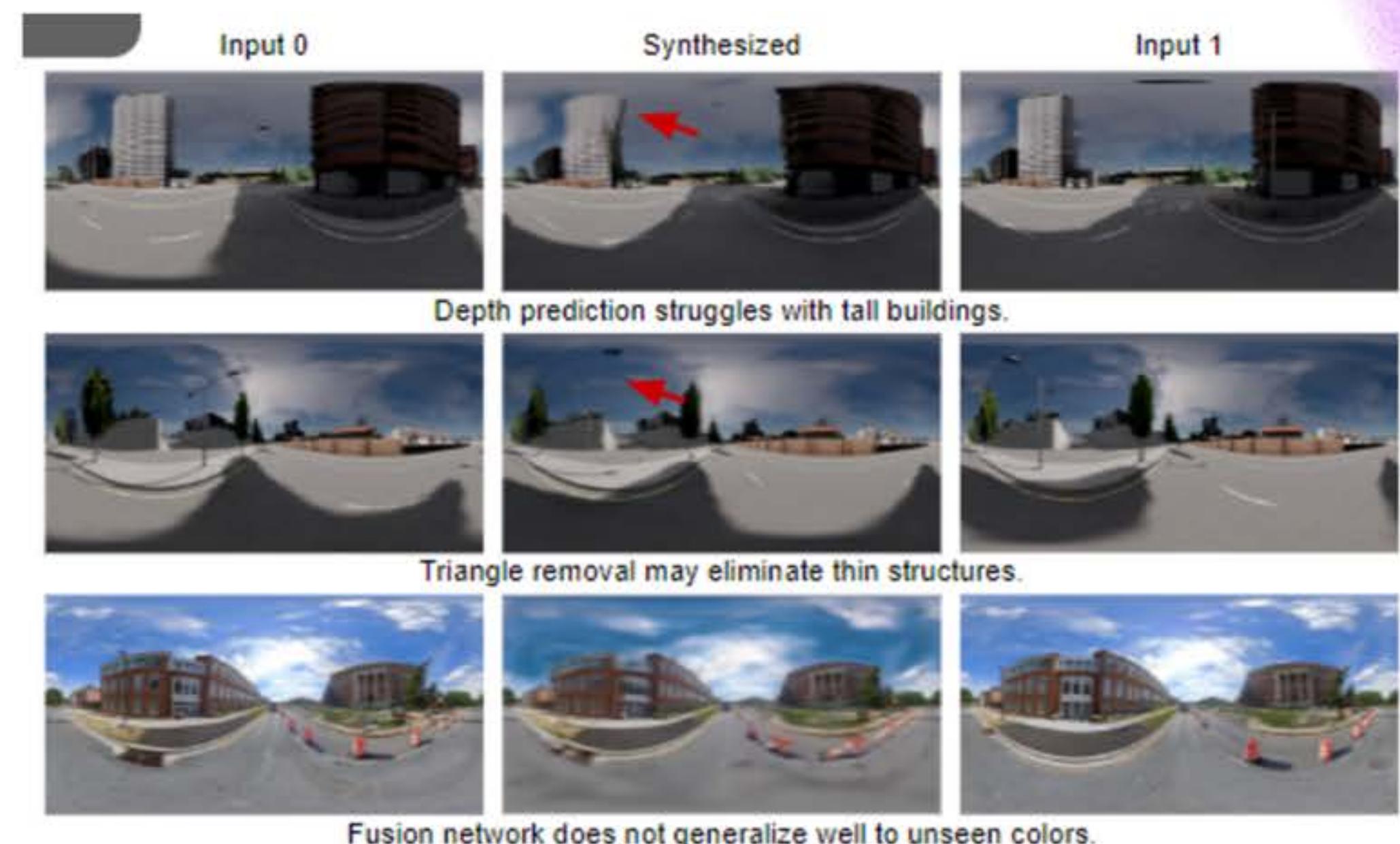
Visualising AI Methods in XR



Generating Training Data for XR

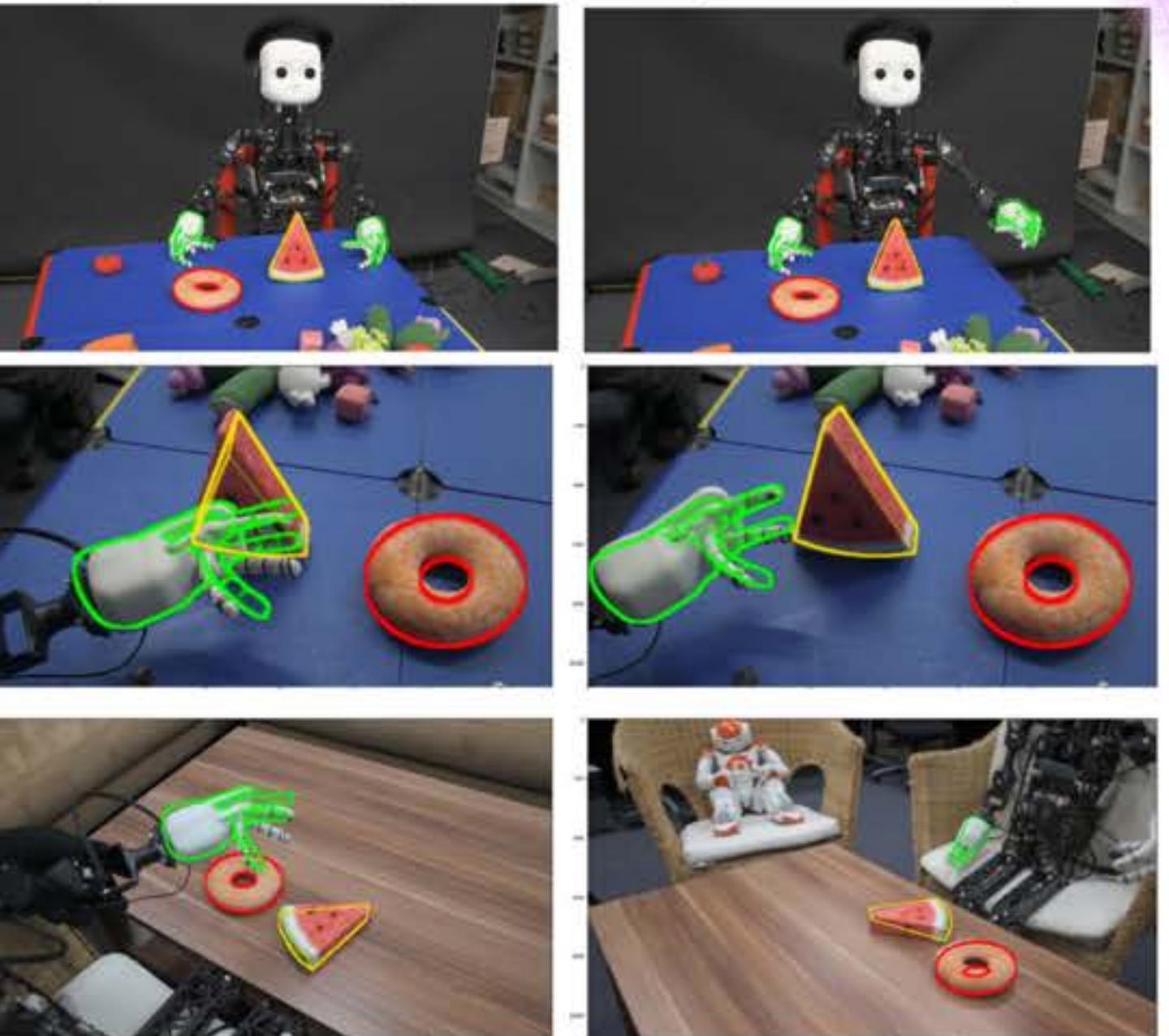
# Visualizing AI Methods in XR

- **Immersive Visualization :**  
Using XR for immersive visualization of complex AI models and data.
- Example: VR environments for visualizing neural networks.



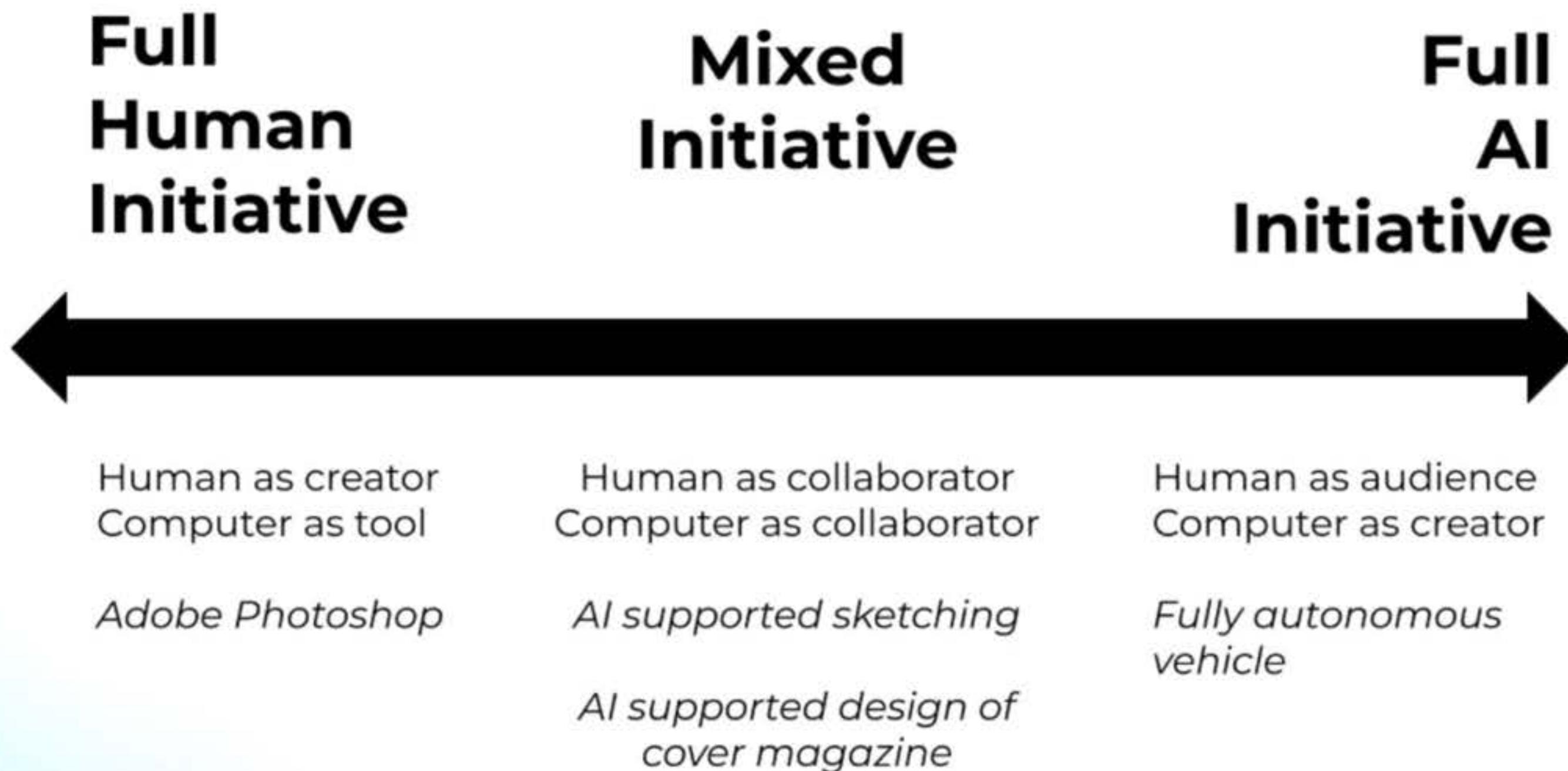
# Generating Training Data for AI

- **Synthetic Data Generation:** AI-based XR tools for creating synthetic datasets for training AI models.
- Example:
  - VR simulations for generating labelled data for autonomous driving.
- Habitat - AI



# Designing AI for XR

# Designing AI4XR



# Designing AI4XR

## # USER AUTONOMY

Design for the appropriate level of user autonomy



## # DATA & MODEL ALIGNMENT

Align AI with real-world behaviors



## # EVOLVING SAFETY

Treat safety as an evolving endeavor



## # ADAPT WITH FEEDBACK

Adapt AI with user feedback



## # HELPFUL AI

Create helpful AI that enhances work and play

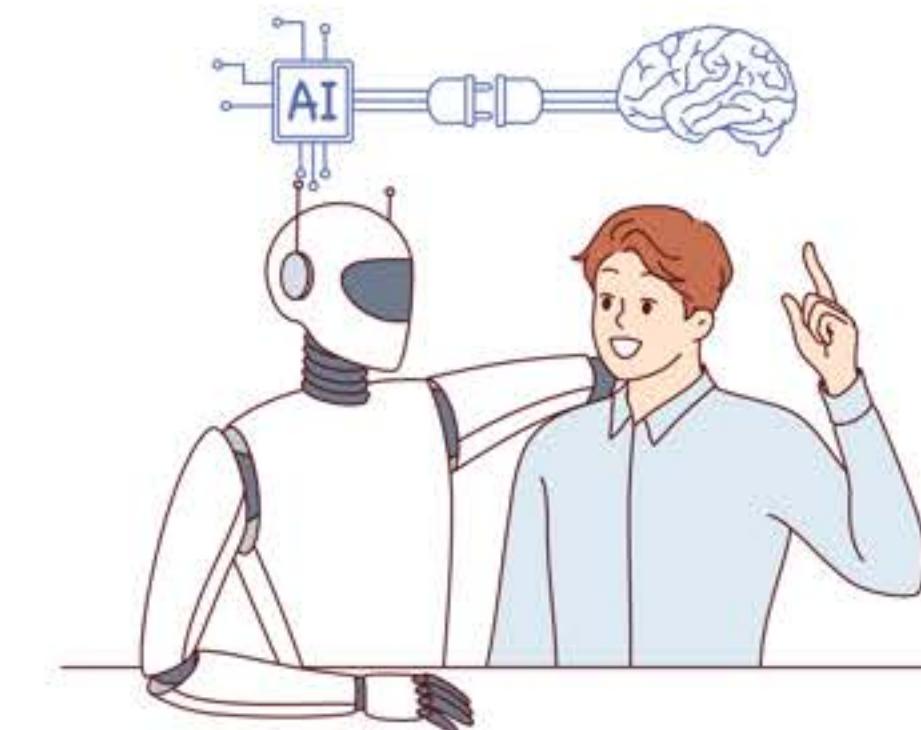


# Automation



Replace Human  
Intelligence with Artificial  
Intelligence

# Collaboration

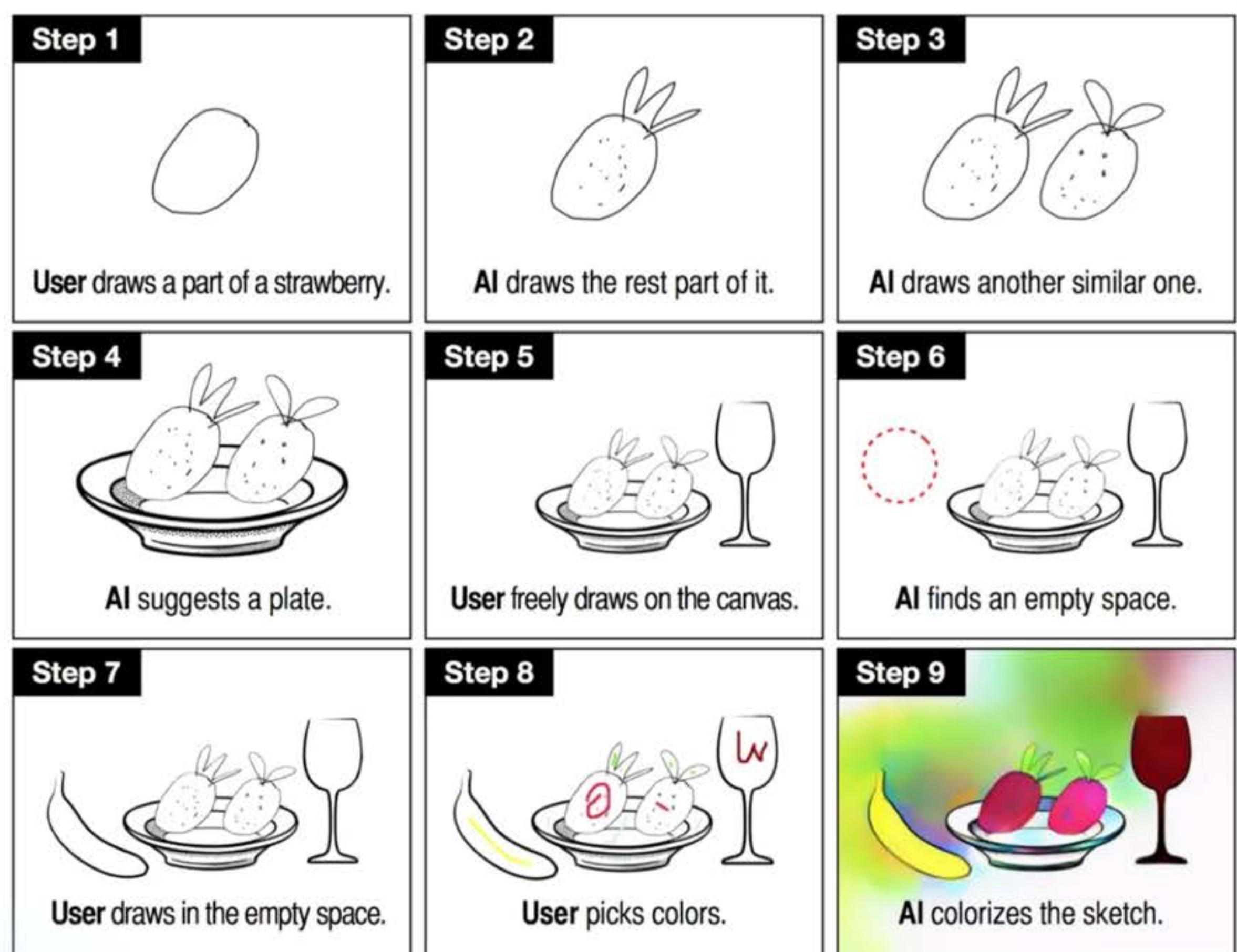


Human Intelligence with  
Artificial Intelligence

# Augmentation



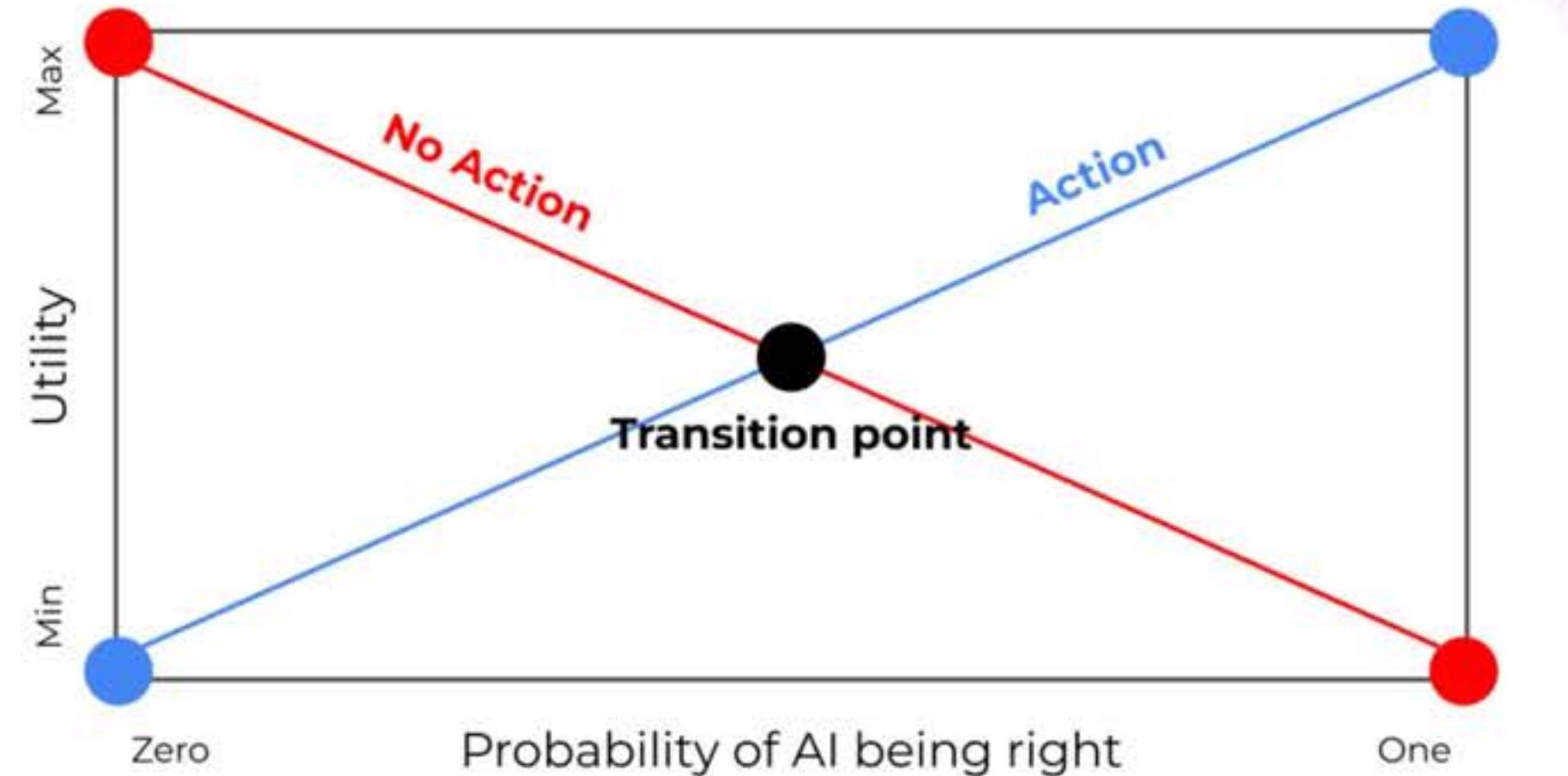
Augment Human  
Intelligence with Artificial  
Intelligence



An example of a mixed-initiative design process for the task of sketching.

# Practical guidance

- a) The AI takes an action AND is correct  
(the AI schedules a meeting and the email actually contained a meeting request).
- b) The AI takes an action BUT is wrong  
(the AI schedules a meeting but there was no meeting request in the email).
- c) The AI takes no action AND is correct  
(AI does not schedule a meeting and there was no meeting request in the email).
- (d) The AI takes no action BUT is wrong  
(AI does not schedule a meeting but there was a meeting request in the email).



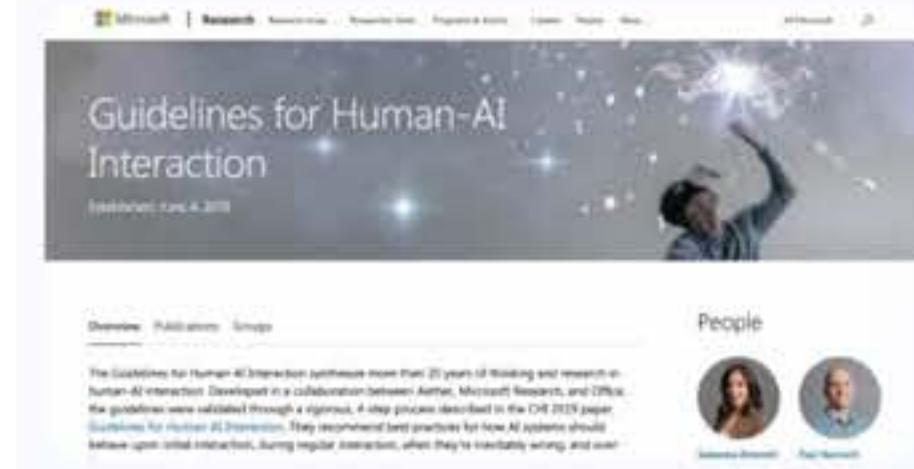
Mixed-initiative: If the probability of the AI being right is higher than a certain threshold perform an action. If it's lower don't act.

# AI Design Guidelines



## Apple HIG. Machine Learning

Apple guidelines for designing UI and user experience of a machine learning app



## Microsoft guidelines for Human-AI interaction

Best practices for how AI systems should behave upon initial interaction, during regular interaction, when they're inevitably wrong, and over time



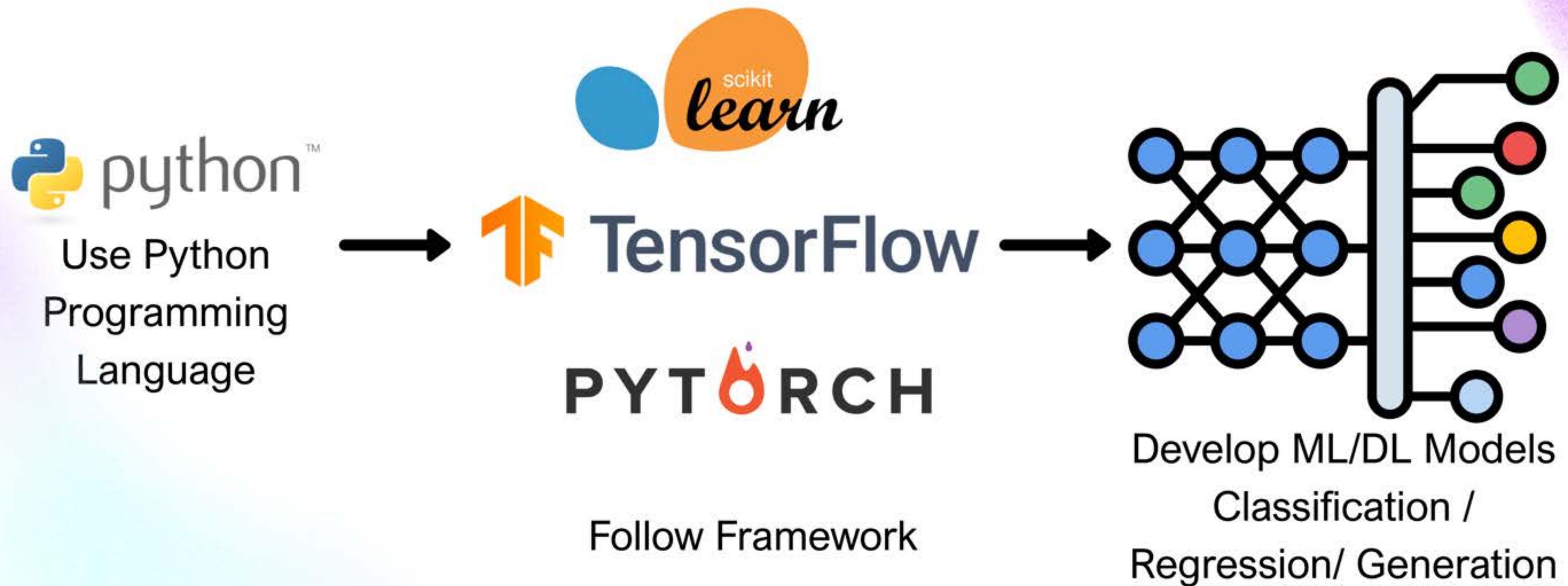
## People + AI Research

People + AI Research (PAIR) is a multidisciplinary team at Google that explores the human side of AI by doing fundamental research, building tools, creating design frameworks, and working with diverse communities. We believe that for machine learning to achieve its positive potential, it needs to be participatory, involving the communities it affects and guided by a diverse set of citizens, policymakers, activists, artists, and more.

# Tools and Frameworks for AI for XR

# Programming - AI Tools

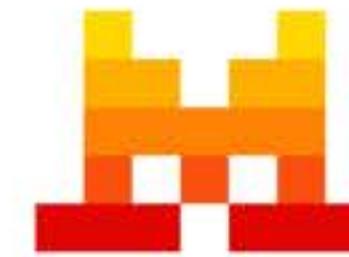
## Machine and Deep Learning Frameworks



# Generative AI Tools



**OpenAI**



**Mistral AI**



**Midjourney**

★ **Gemini**



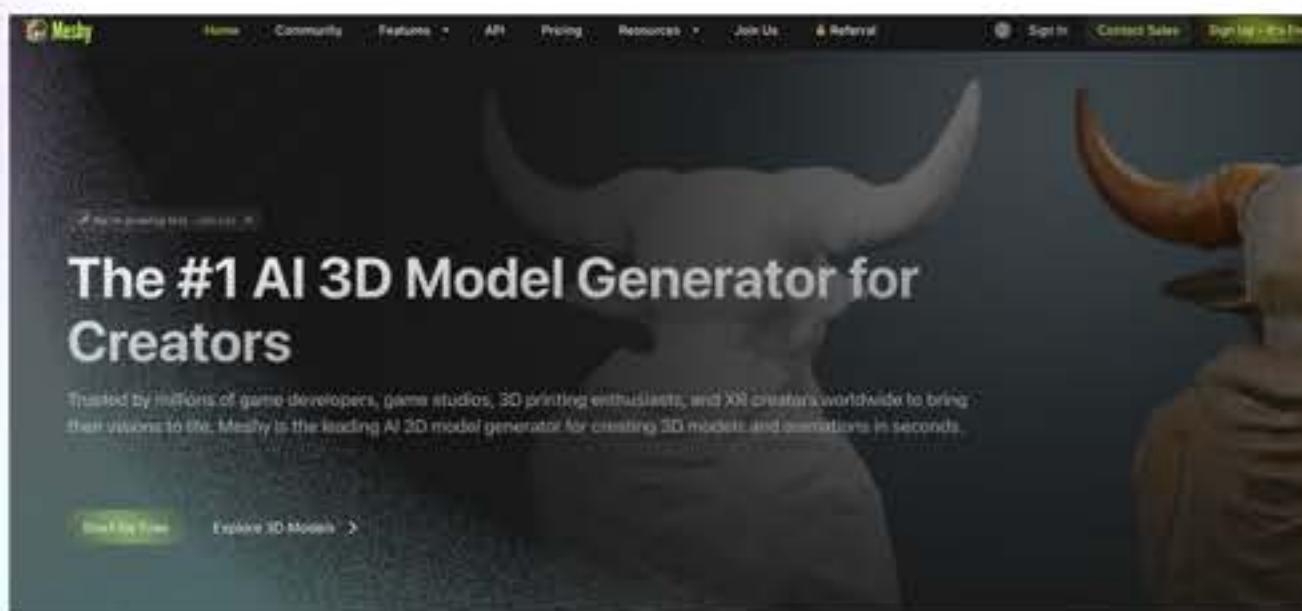
 **Claude**



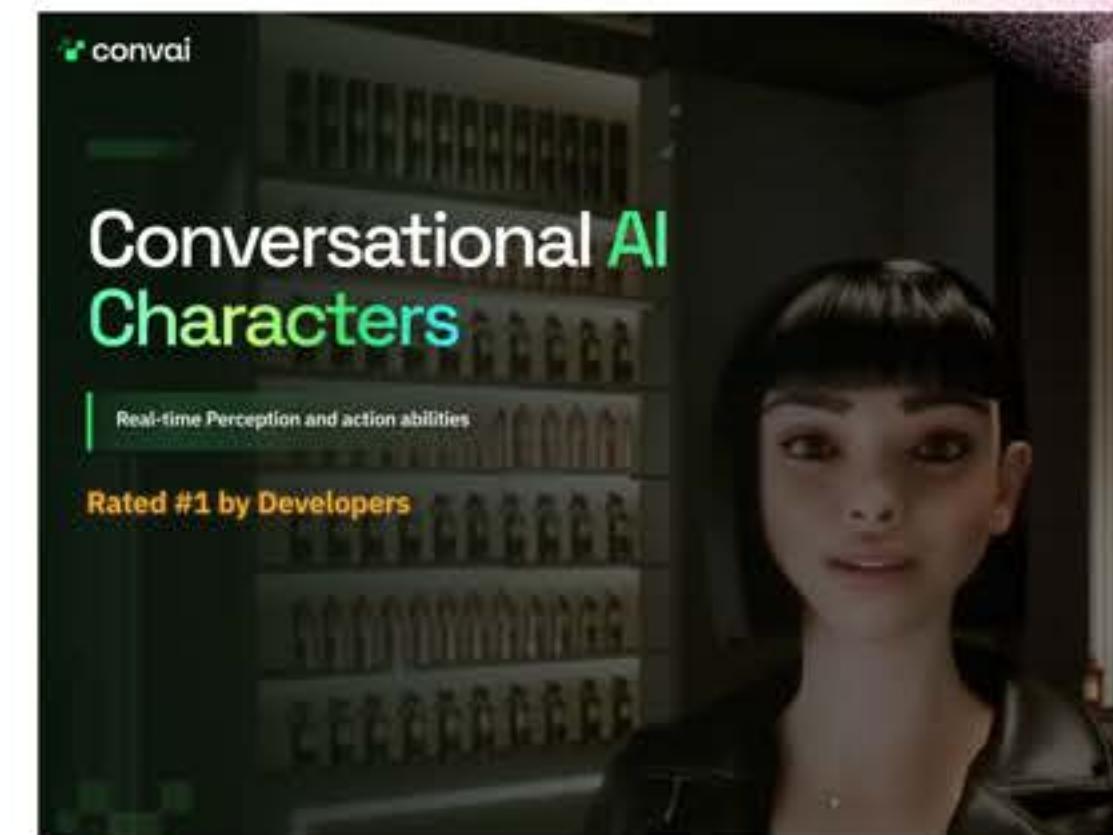
# Generative AI Tools for XR



Text to SkyBox Generation



Text to 3D Models



Conversational Agents



Text to Music

# Finding AI Models for your Applications

 **Hugging Face**

[Models](#) [Datasets](#) [Spaces](#) [Community](#) [Docs](#) [Pricing](#) [Log In](#) [Sign Up](#)



## The AI community building the future.

The platform where the machine learning community collaborates on models, datasets, and applications.

[Explore AI Apps](#) or [Browse 1M+ models](#)

Tasks   Libraries   Datasets   Languages   Licenses   Other

Filter Tasks by name

Multimodal

- Text-to-Image
- Image-to-Text
- Text-to-Video
- Visual Question Answering
- Document Question Answering
- Graph Machine Learning

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Image-to-Image
- Unconditional Image Generation
- Video Classification
- Zero-Shot Image Classification

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Cooersational
- Text Generation
- Text2Text Generation
- Sentence Similarity

Audio

- Text-to-Speech
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

Tabular

- Tabular Classification
- Tabular Regression

Reinforcement Learning

Models 469,541 [Filter by name](#)

- meta-llama/llama-2-70b  
Text Generation • Updated 4 days ago • ± 25.2k • ▲ 64
- stabilityai/stable-diffusion-xl-base-0.9  
Updated 6 days ago • ± 2.01k • ▲ 393
- openchat/openchat  
Text Generation • Updated 2 days ago • ± 1.3k • ▲ 136
- illyasviel/ControlNet-v1-1  
Updated Apr 26 • □ 1.87k
- cerspense/zeroscope\_v2\_XL  
Updated 3 days ago • ± 2.66k • ▲ 334
- meta-llama/llama-2-13b  
Text Generation • Updated 4 days ago • ± 3.32k • ▲ 64
- tiiuae/falcon-40b-instruct  
Text Generation • Updated 27 days ago • ± 2.88k • ▲ 899
- WizardLM/WizardCoder-15B-V1.0  
Text Generation • Updated 3 days ago • ± 12.5k • □ 332
- CompVis/stable-diffusion-v1-4  
Text-to-Image • Updated about 17 hours ago • ± 443k • ▲ 5.72k
- stabilityai/stable-diffusion-2-1  
Text-to-Image • Updated about 17 hours ago • ± 782k • ▲ 2.81k
- Salesforce/xgen-7b-8k-instruct  
Text Generation • Updated 4 days ago • ± 6.18k • □ 57

# Teachable Machine

teachablemachine.withgoogle.com

About FAQ Get Started

## Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

Get Started

TensorFlow.js p5.js Coral Node.js

### What is Teachable Machine?

Class 1

Add Pose Samples:

Webcam Upload

Class 2

Add Pose Samples:

Webcam Upload

Training

Train Model

Advanced

Epochs: 50

Batch Size: 16

Learning Rate: 0.0001

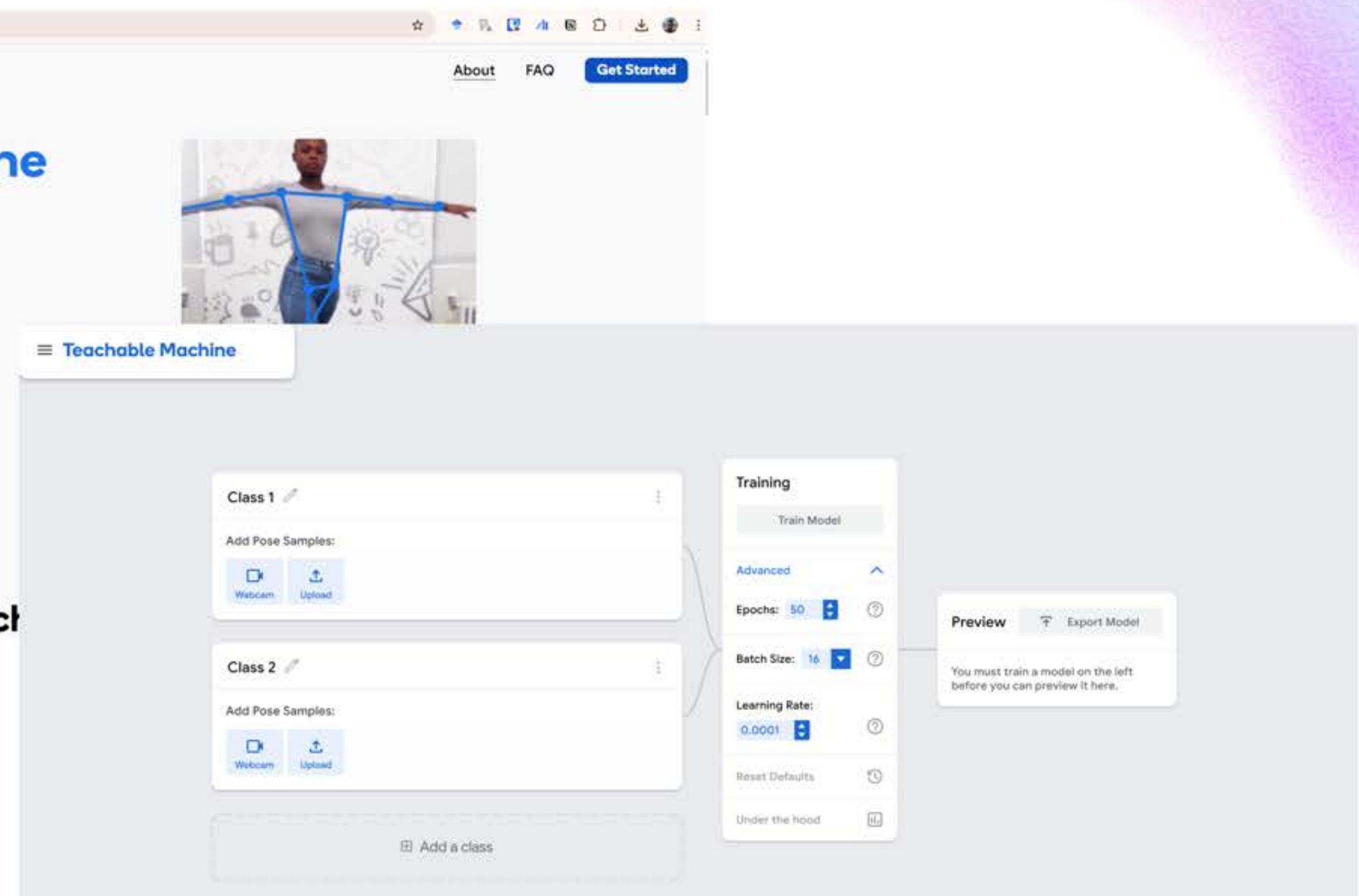
Reset Defaults

Under the hood

Preview Export Model

You must train a model on the left before you can preview it here.

Add a class



# Google AI Studio

Google AI Studio

Chat Stream Generate Media Build History

Chat Prompt

Welcome to AI Studio

Generate a scavenger hunt for street food around the city of Seoul, Korea. →

Run

What's new

- URL context tool Fetch information from web links
- Native speech generation Generate high quality text to speech with Gemini
- Live audio-to-audio dialog Try Gemini's natural, real-time dialog with audio and video inputs
- Native image generation Interleaved text-and-image generation with Gemini 2.0 Flash

Google AI models may make mistakes, so double-check outputs.

Run settings

Gemini 2.5 Pro

Token count 0 / 1,048,576

Temperature 1

Media Resolution Default

Thinking

Thinking mode

Set thinking budget

Tools

Structured output

Code execution

Function calling

Grounding with Google Search

URL context

# Local LLM



## LM Studio

- User-friendly desktop app for local LLM inference.
- Supports GGUF, GPTQ, and Hugging Face models.
- Simple GUI to manage models, sessions, and APIs.



- Command-line tool for running local LLMs efficiently.
- Supports various open models (LLaMA, Gemma, Mistral).
- Strong CLI-focused, suitable for developers.

# Unity AI-XR Tutorial





# Unity AI

- Unity AI is an integrated artificial intelligence platform built into Unity Editor
- Designed to help developers create smarter games and applications
- Three main components:
  - a. **AI Assistant** - Code generation and development help
  - b. **AI Generators** - Asset creation (sprites, textures, animations, sounds)
  - c. **Inference Engine** - Run AI models locally in your projects

# The Three Pillars of Unity AI

## 1. AI Assistant



Your coding companion

- Answers questions about Unity development
- Generates code snippets
- Helps with debugging and optimization

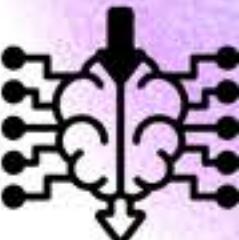
## 2. AI Generators



Creative asset production

- Generate sprites and images from text
- Create textures and materials
- Produce animations and sound effects

## 3. Inference Engine



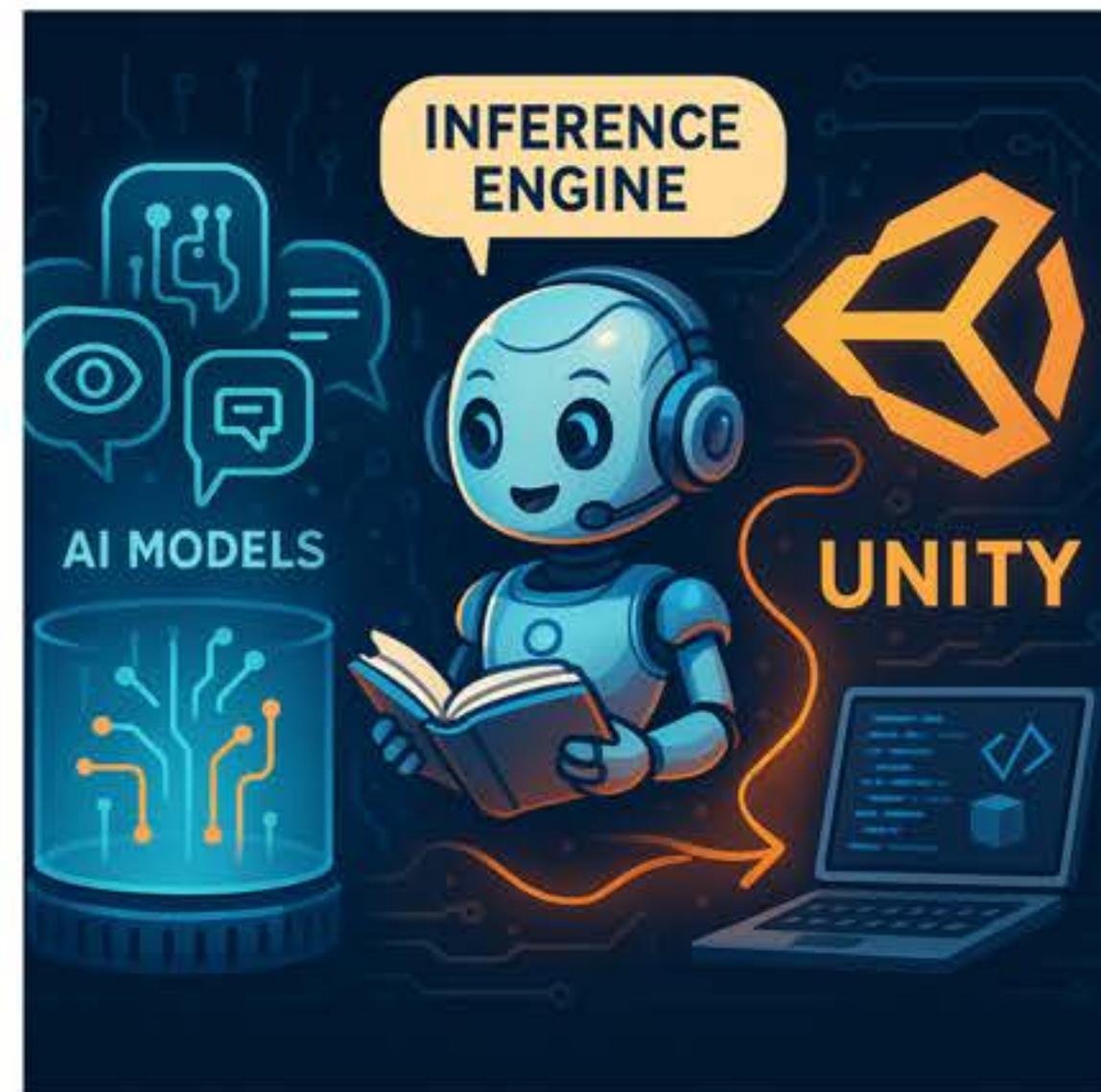
The powerhouse

- Run pre-trained AI models locally
- Real-time AI in your games
- Works on all Unity platforms

# Unity AI Inference Engine

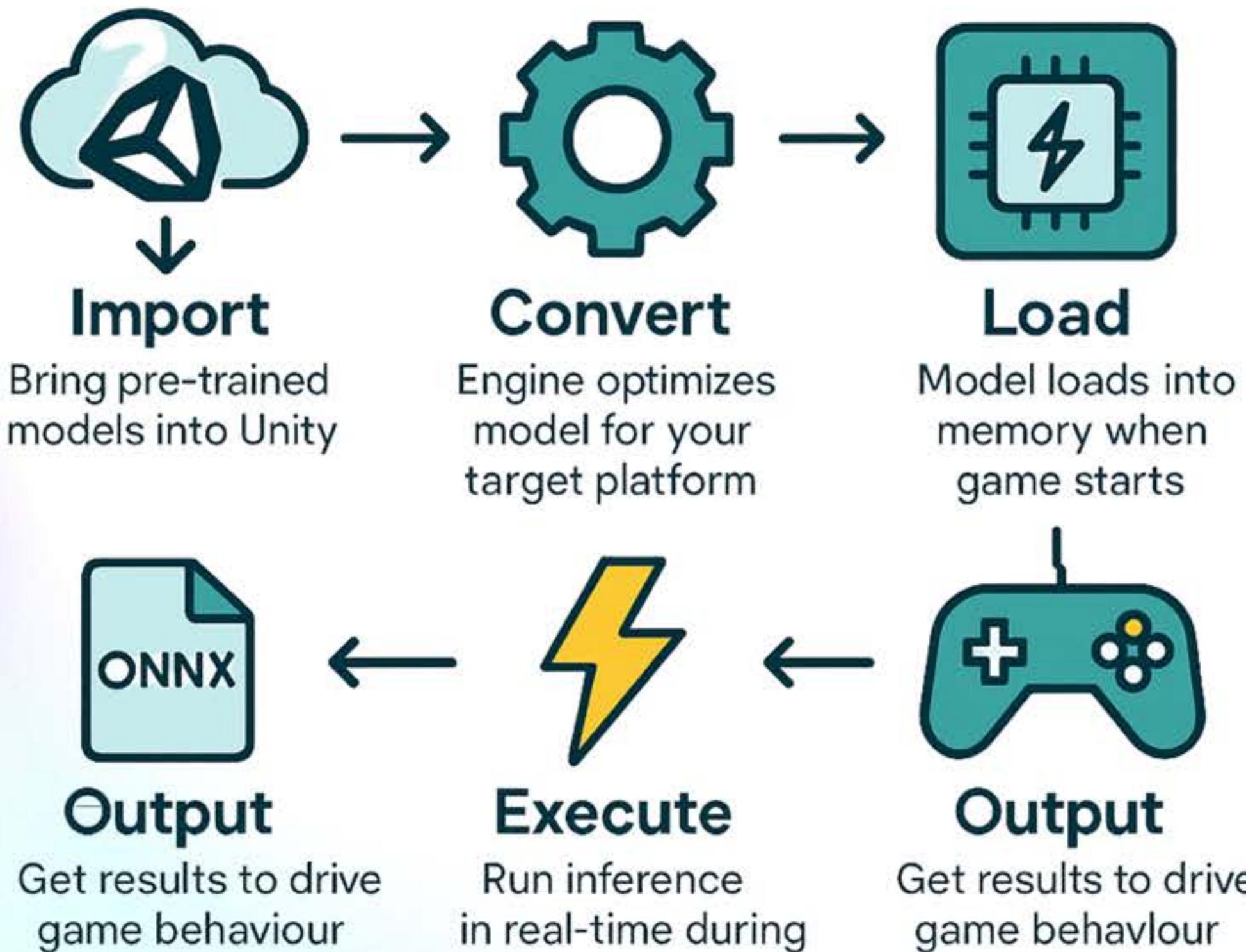


- Neural network inference library that runs AI models locally
- Key Features:
  - Runs trained models in real-time



“A translator that takes AI models and makes them speak Unity's language”

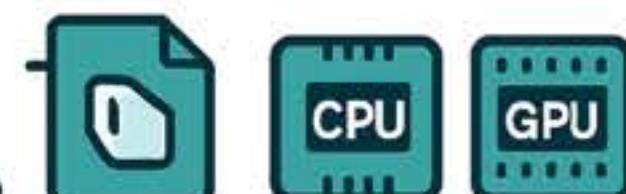
# How Inference Engine Works



1. Throttle inference (every N frames) on WebGL/mobile to keep FPS > 60.
2. Use quantised or MobileNet-style architectures for AR on phones.
3. Pre-warm the worker in a loading scene to avoid first-frame hitch.
4. Label outputs clearly in your training graph; Sentis exposes them by name.

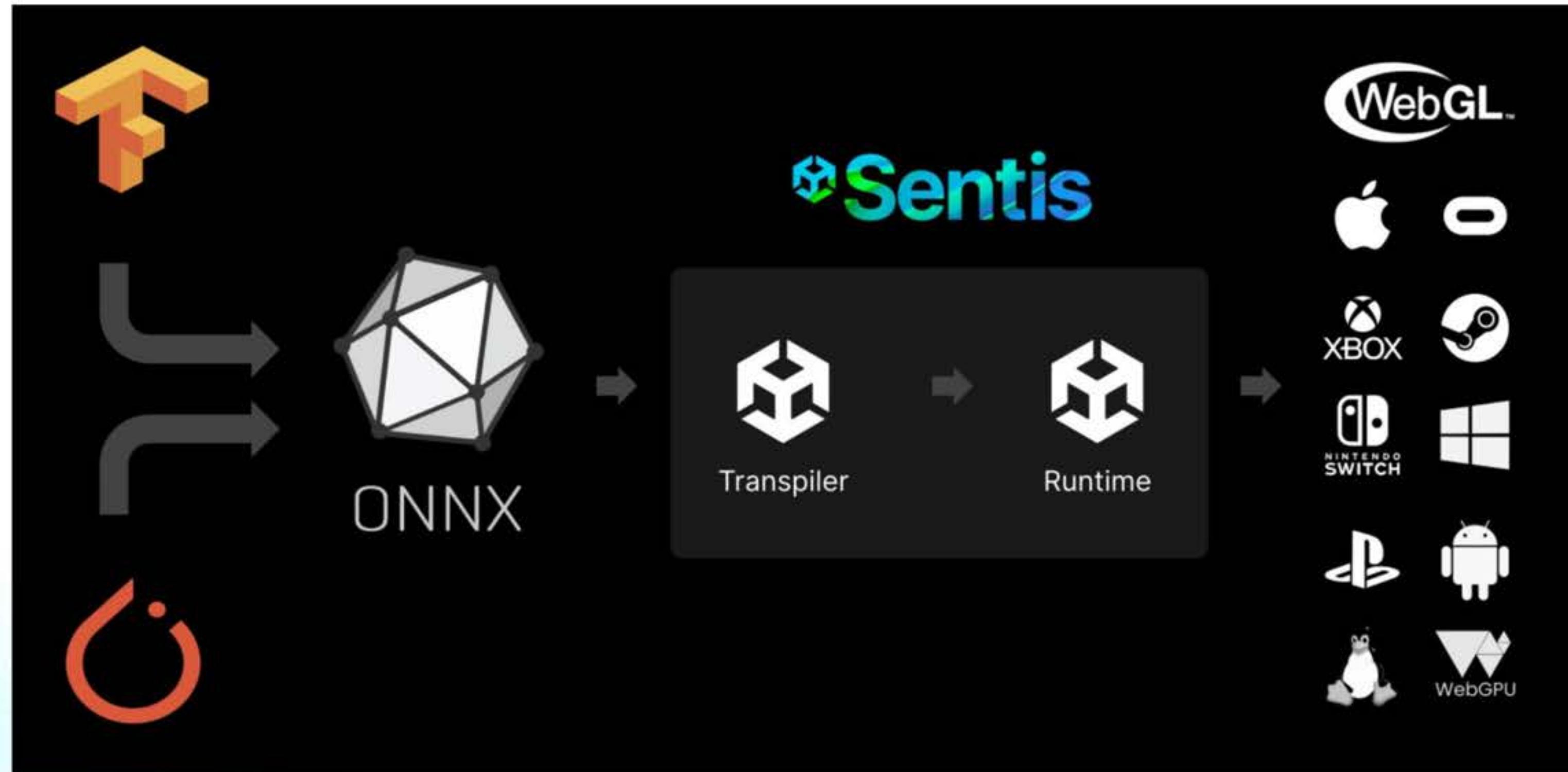
## Model Support:

ONNX format (industry standard)  
CPU and GPU execution



CPU and GPU execution

# Integrating AI Models in Unity



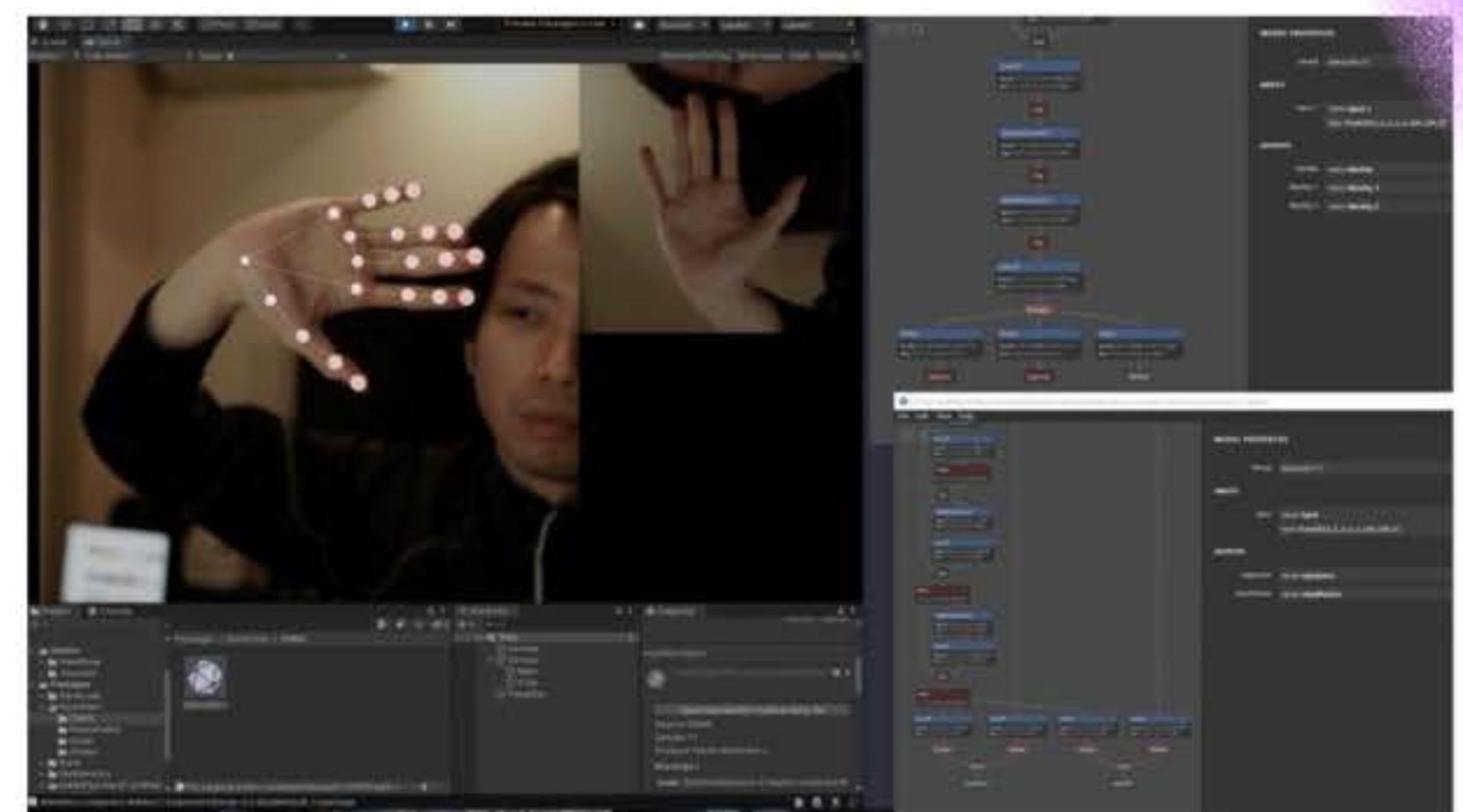
# Setting Up Your First Model

## Step 1: Get a Model

- Download from Hugging Face, Kaggle Models, PyTorch Hub
- Or train your own and export to ONNX

## Step 2: Import to Unity

- Add model file to your Unity project
- Use Package Manager to install Inference Engine
- Create model asset in Unity



# Setting Up Your First Model

## Step 3: Basic Setup

```
void Start()
{
    // Load the model into a runtime format
    _runtimeModel = ModelLoader.Load(modelAsset);

    // Create an inference engine; picks GPU if available, falls back to CPU
    _worker = WorkerFactory.CreateWorker(WorkerFactory.Type.Auto, _runtimeModel);
}
```

# Setting Up Your First Model

## Step 4: Running Inference

```
void Update()
{
    // Example input: 1 batch, 1x1 image with 10 channels
    var inputTensor = new Tensor(1, 1, 1, 10); // adjust sizes to match your model

    // Fill the tensor with sample data (e.g., random or actual inputs)
    for (int i = 0; i < 10; i++)
        inputTensor[0, 0, 0, i] = Random.value;

    // Run inference
    _worker.Execute(inputTensor);

    // Retrieve output (assuming a single output)
    var outputTensor = _worker.PeekOutput();

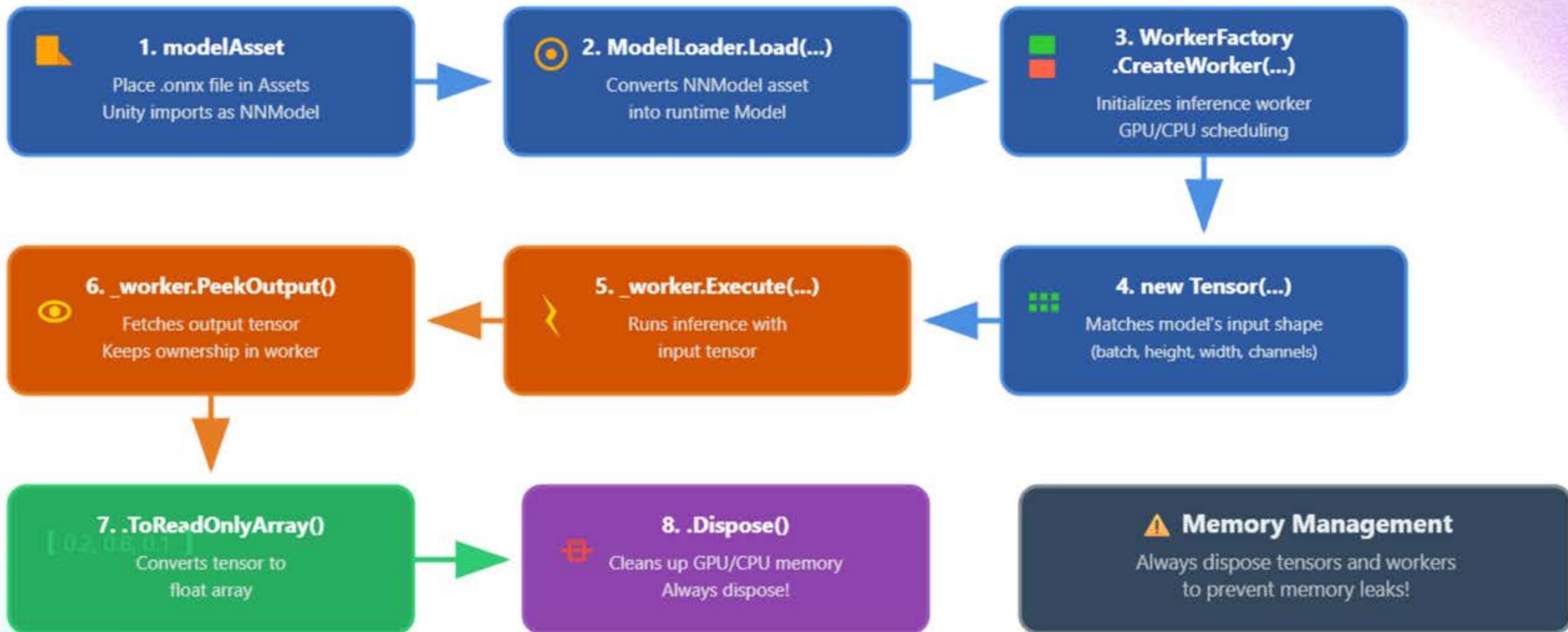
    // Convert output to float array and log
    float[] outputArray = outputTensor.ToReadOnlyArray();
    Debug.Log("Model output: " + string.Join(", ", outputArray));

    // Clean up tensors
    inputTensor.Dispose();
    outputTensor.Dispose();
}
```

# Setting Up Your First Model

## Step 5: Dispose

```
0 references
void OnDestroy()
{
    // Dispose the worker to free resources
    _worker.Dispose();
}
```





# Unity AI Performance Tips

1

## Reuse Workers

Create once, use multiple times

Don't create new workers for each inference

2

## Batch Processing

Process multiple inputs together

Better GPU utilization and throughput



3

## Use GPU Backend



Leverage GPU for complex models

Especially for image/vision models



4

## Async Execution



Don't block main thread

Use coroutines or async/await patterns



**Key Point:** Always dispose workers and tensors to prevent memory leaks!

Memory management is critical!

# ONNX Model Database

Hugging Face

Models Datasets Spaces Posts Docs Solutions Pricing

Tasks Libraries 1 Datasets Languages Licenses Other

Filter Libraries by name

PyTorch TensorFlow JAX Transformers

TensorBoard Safetensors Diffusers PEFT

Stable-Baselines3 ONNX ML-Agents

Sentence Transformers GGUF Flair Timm

Sample Factory Keras SetFit Adapter Transformers

Transformers.js spaCy ESPnet fastai

Core ML NeMo Rust Joblib fastText

Scikit-learn OpenCLIP speechbrain BERTopic

PaddlePaddle OpenVINO Fairseq Graphcore

MLX TF Lite Asteroid Stanza allenNLP

PaddleNLP SpanMarker Habana TensorFlowTTS

pyannote.audio Pythaë K2 Unity Sentis docTR

MindSpore

Models 12 Filter by name

new Full-text search Sort: Trending

<b>unity/sentis-tiny-stories</b> Updated 3 days ago	<b>unity/sentis-yolotinyv7</b> Updated 3 days ago
<b>unity/sentis-MiDaS</b> Updated 3 days ago	<b>unity/sentis-mobilenet-v2</b> Updated 3 days ago
<b>unity/sentis-othello</b> Updated 3 days ago	<b>unity/sentis-MNIST-12</b> Updated 3 days ago
<b>unity/sentis-whisper-tiny</b> Updated 3 days ago	<b>unity/sentis-neural-cellular-automata</b> Updated 3 days ago
<b>unity/sentis-blaze-face</b> Updated about 16 hours ago	<b>unity/sentis-audio-frequency-to-16khz</b> Updated 3 days ago
<b>unity/sentis-MiniLM-v6</b> Updated 3 days ago	<b>julienkay/sentis-MiDaS</b> Updated 8 minutes ago

# Implementing Pose Detection Logic

## 1. Capture and Process Image:

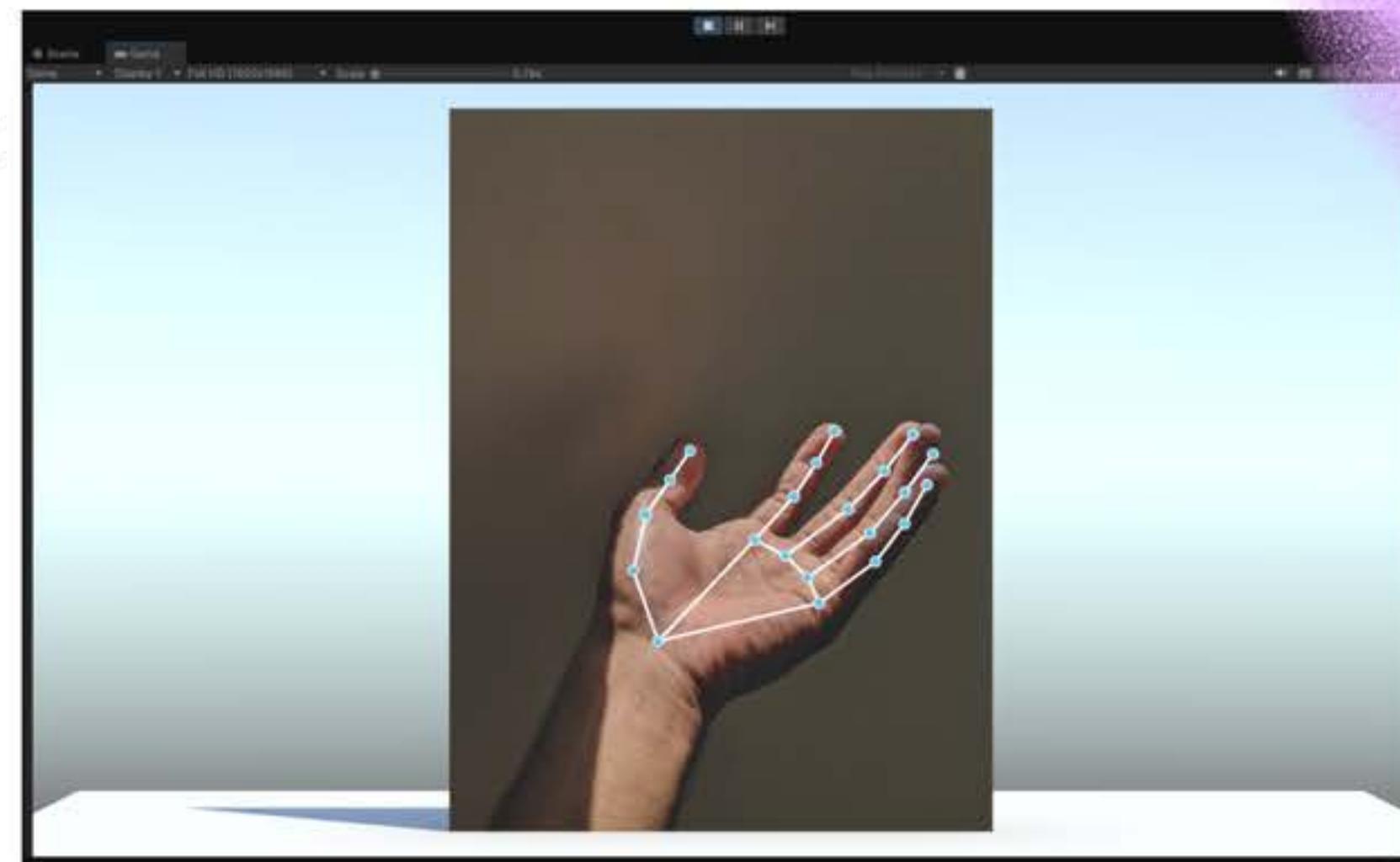
- Add a script to handle image input from the UI.
- Resize the image to match the model input size.

## 2. Run Inference:

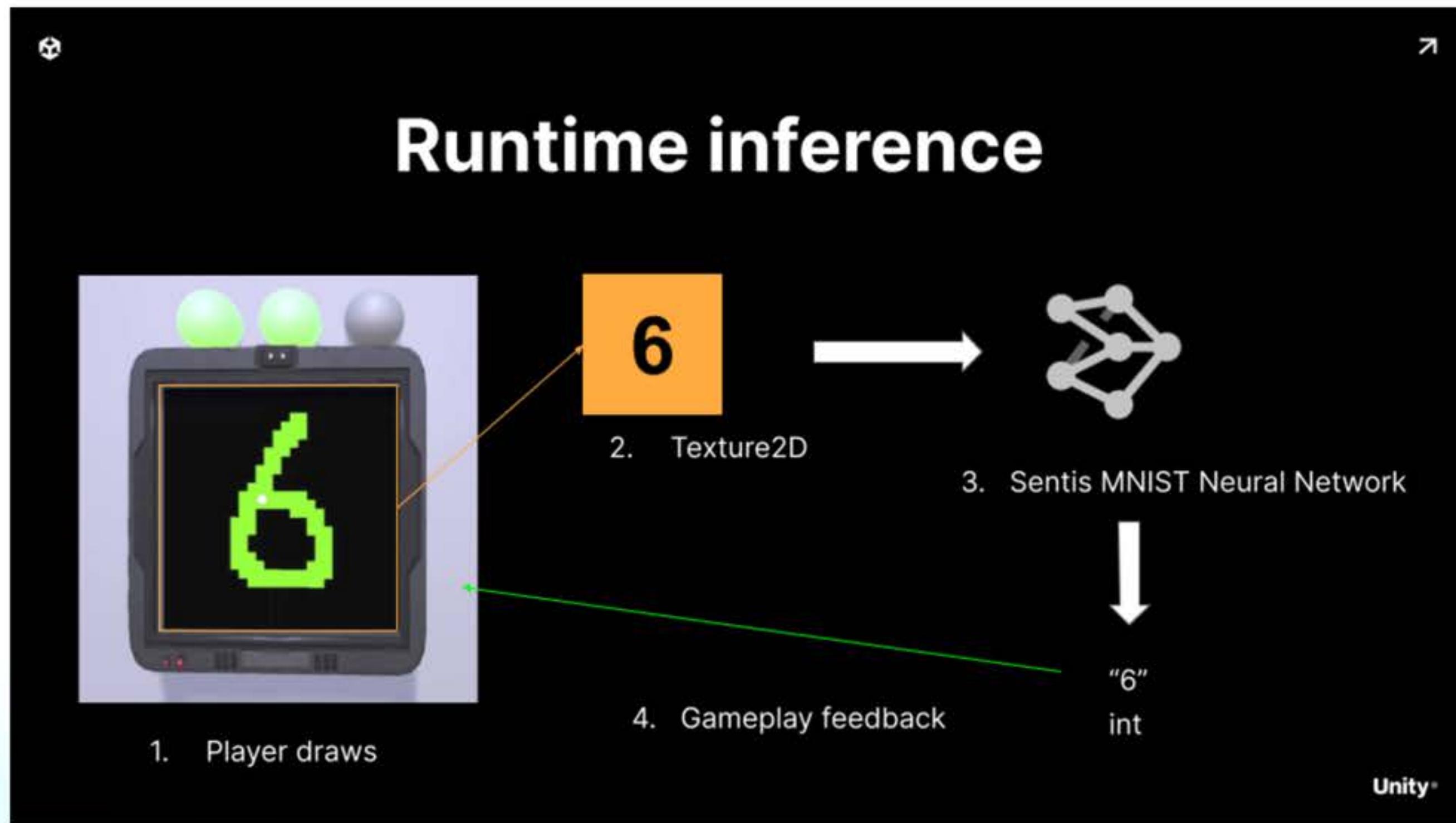
- Pass the processed image to the Sentis model.
- Retrieve output tensor containing hand positions.

## 3. Visualize Results:

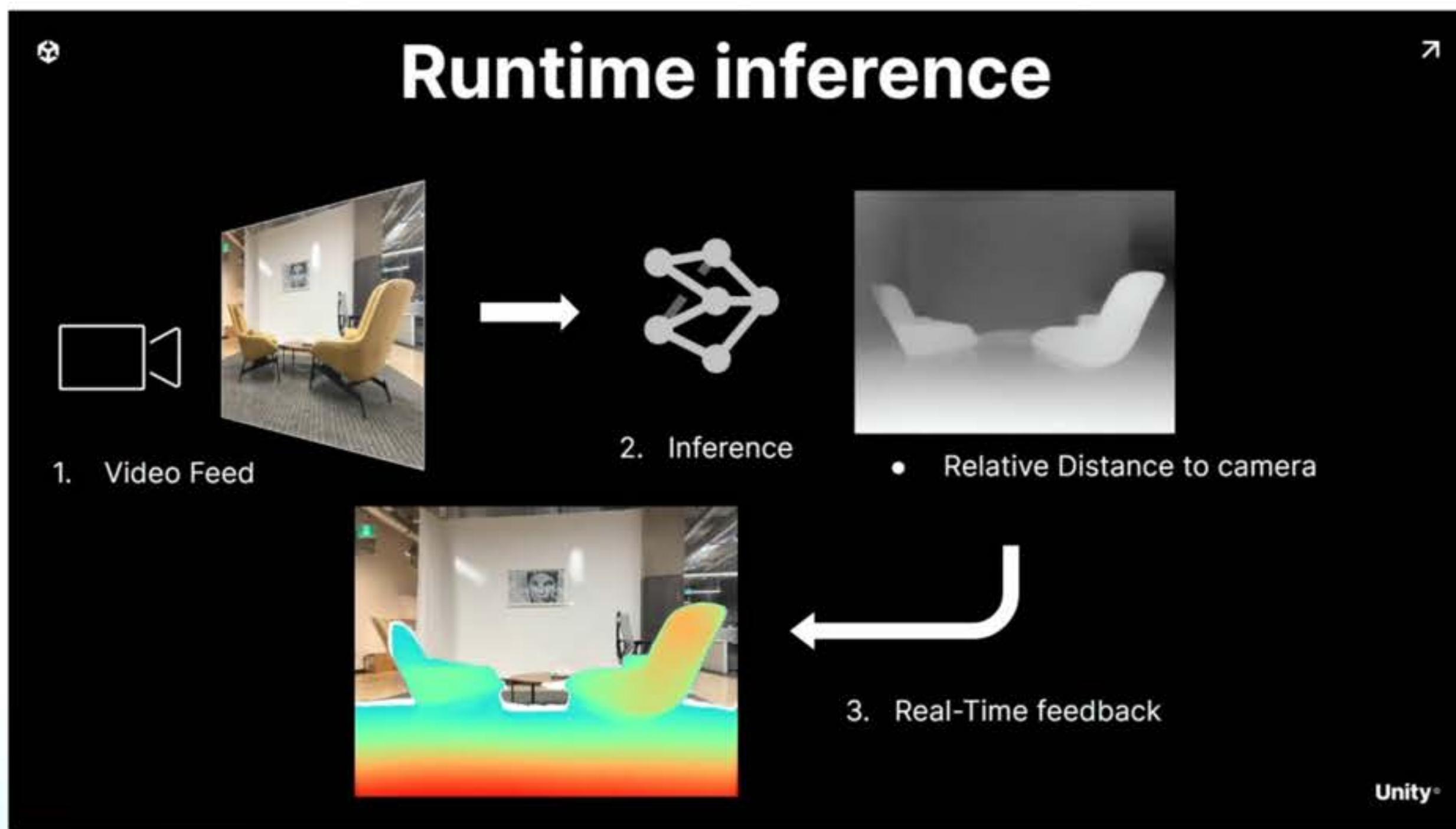
- Draw bounding boxes around detected hands.
- Display the result on the UI panel



# Image Classification



# Depth Estimation



# Board Game

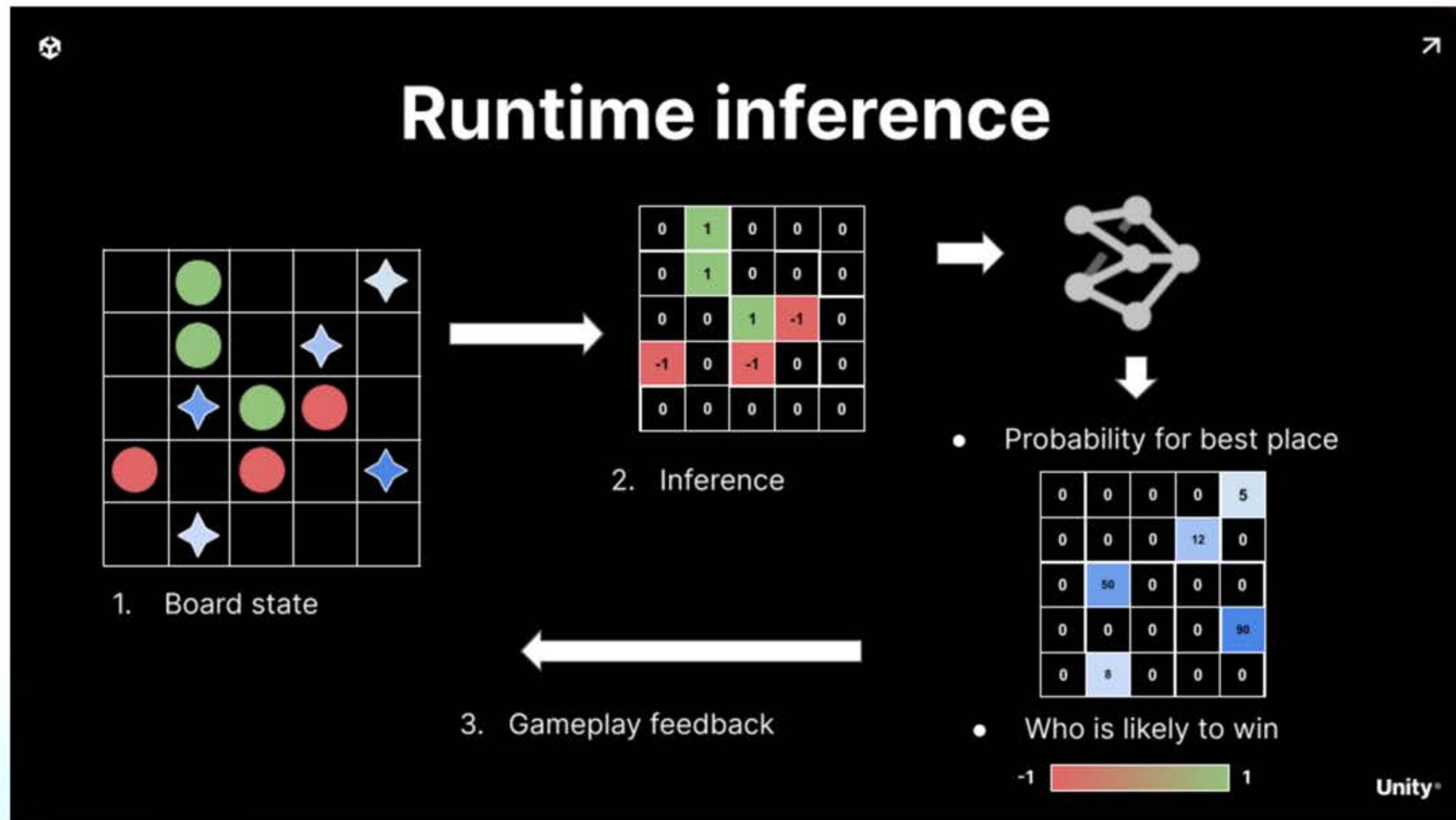
The screenshot shows a 3D Othello board on a wooden surface. A green line highlights the AI difficulty selection area, which includes 'Easy', 'Medium', and 'Best' options and a portrait of a character. A yellow line highlights the board itself, showing pieces in play. A red line highlights the text 'Aha! I take the lead!' at the bottom.

- AI has different difficulty modes
- Show likely next moves + win rate
- Quotes based on Game-State

Aha! I take the lead!

Unity®

# Board Game



# LLM for Unity

huggingface.co/lmstudio-community/gemma-3n-E4B-it-text-GGUF

Hugging Face Search models, datasets, users...

lmstudio-community gemma-3n-E4B-it-text-GGUF Follow LM Studio Community 2.3K

GGUF conversational Llama: gemma

Model card Files and versions Community Edit model card Deploy Use this model

Community Model > gemma-3n-E4B-it-text by Google

LM Studio Community models highlights program. Highlighting new & noteworthy models by the community. Join the conversation on [Discord](#).

Model creator: google  
Original model: [gemma-3n-E4B-it](#)  
GGUF quantization: provided by LM Studio team using [llama.cpp](#)  
Multi-modal quants coming soon.

Special thanks

Special thanks to [Georgi Gerganov](#) and the whole team working on [llama.cpp](#) for making all of this possible.

Disclaimers

LM Studio is not the creator, originator, or owner of any Model featured in the Community Model Program. Each Community Model is created and provided by third parties. LM Studio does not

Downloads last month 19,266

GGUF Model size 6.87B params Architecture gemma3n Chat template

Hardware compatibility RTX 3060 (8 GB) x1

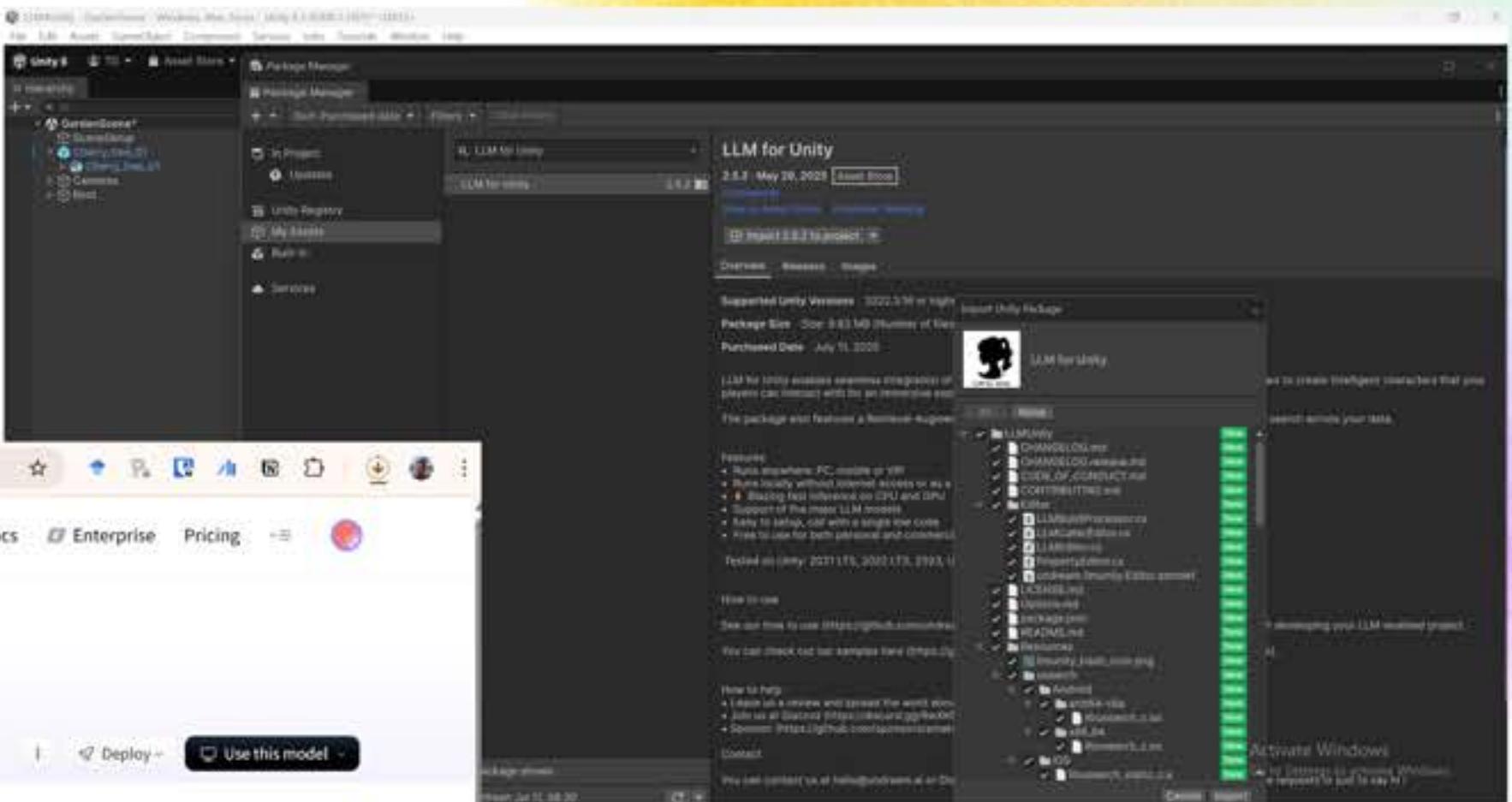
4-bit Q4\_K\_M 4.24 GB

8-bit Q4\_K\_1.57 GB

8-bit Q8\_0 7.15 GB

Inference Providers This model isn't deployed by any Inference Provider. Ask for provider support

Activate Windows Go to Settings to activate Windows





# Best Practices for AI in XR

## 1. Optimize for Performance

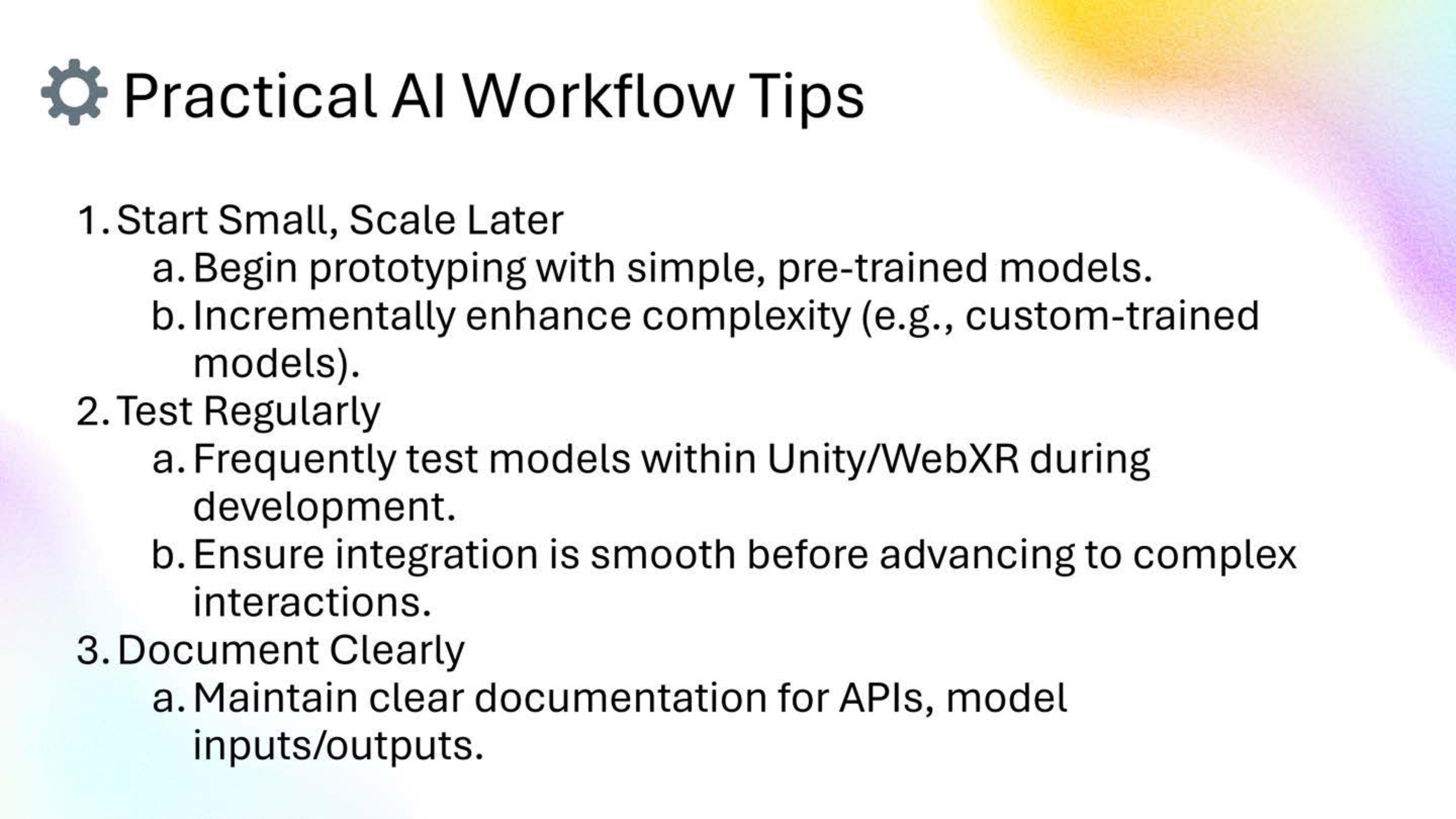
- a. Use lightweight models (e.g., Tiny YOLO, MobileNet, Gemma3n) for AR/Mobile devices.
- b. Limit inference frequency (not every frame—use intervals).

## 2. Prioritize Latency & Responsiveness

- a. Always test response times for cloud APIs.
- b. Use loading indicators or placeholders while AI is processing.

## 3. Design Intuitive Interactions

- a. Ensure XR interactions feel natural (voice, gesture, gaze).
- b. Always provide clear feedback on AI-driven responses.



# Practical AI Workflow Tips

## 1. Start Small, Scale Later

- a. Begin prototyping with simple, pre-trained models.
- b. Incrementally enhance complexity (e.g., custom-trained models).

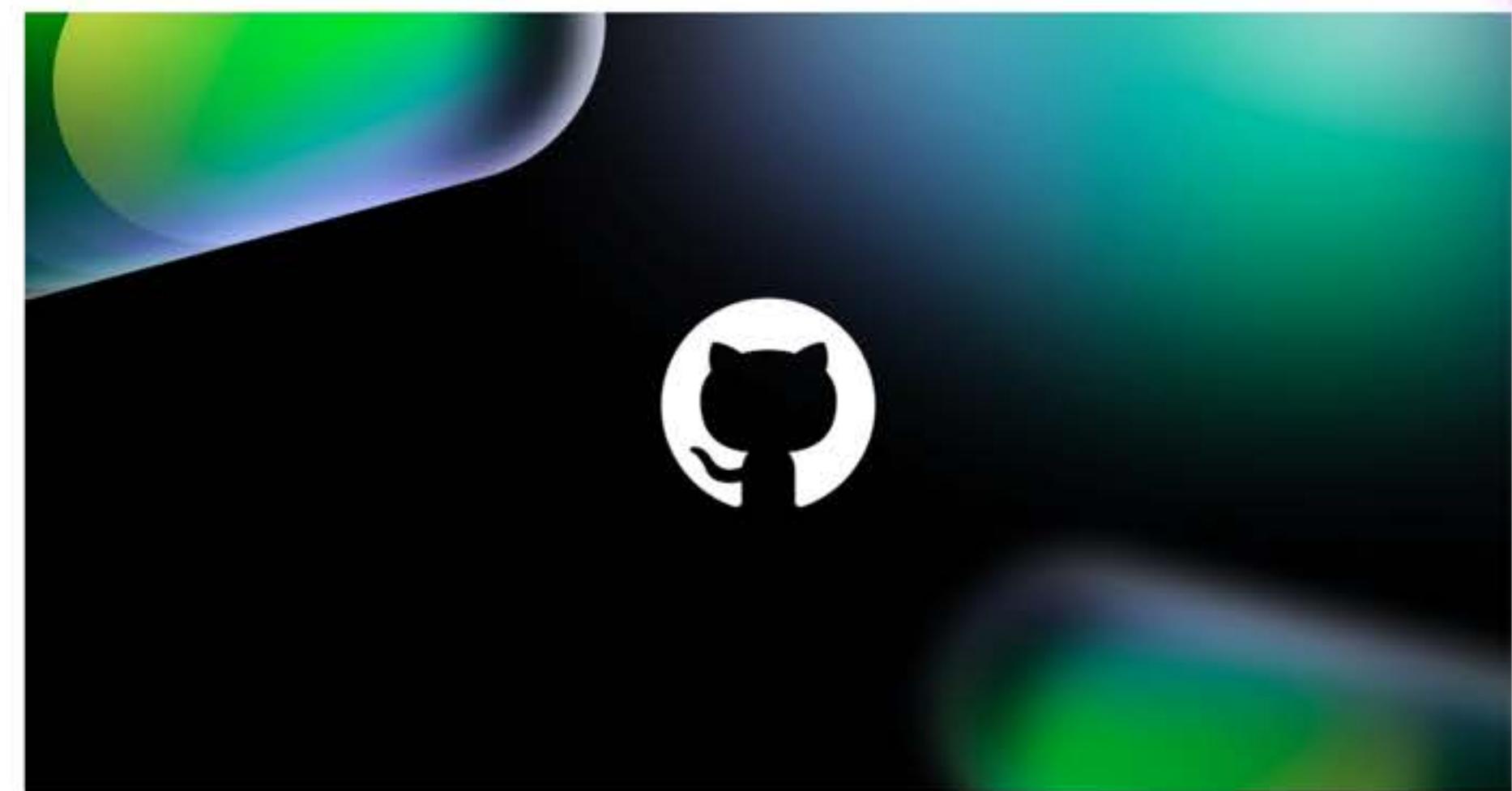
## 2. Test Regularly

- a. Frequently test models within Unity/WebXR during development.
- b. Ensure integration is smooth before advancing to complex interactions.

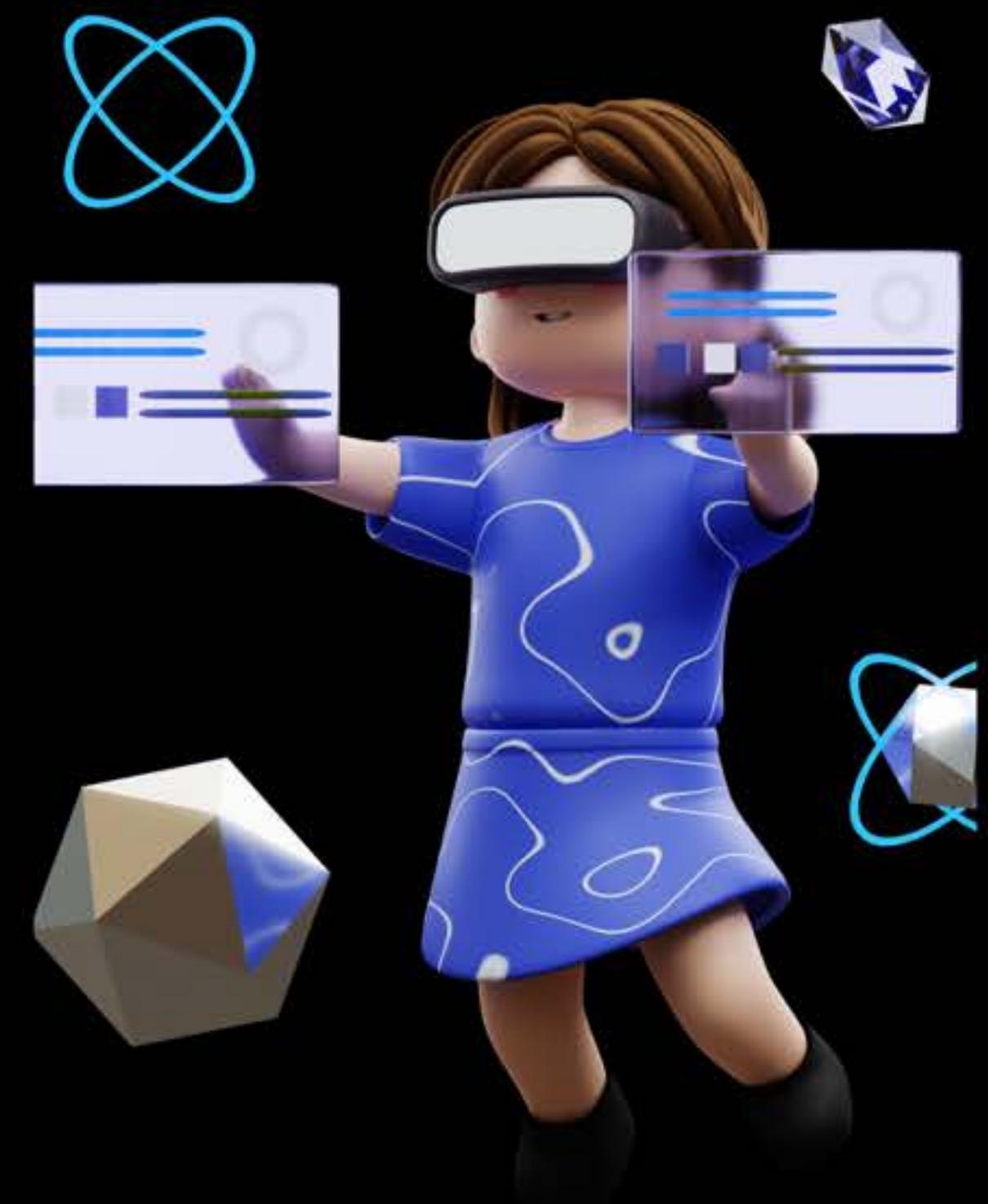
## 3. Document Clearly

- a. Maintain clear documentation for APIs, model inputs/outputs.

# Github Repository



# Summary



# Key Takeaways



**AI as a Game-Changer in XR**



**Unity's AI and open APIs (e.g., Hugging Face, Google AI, Ollama) enable efficient integration of ML models.**



**User-Centric AI Experiences**



**Performance & Latency Matter**





Questions