AI for XR

Crafting Intelligent XR Applications with AI

Tamil Selvan Gunasekaran

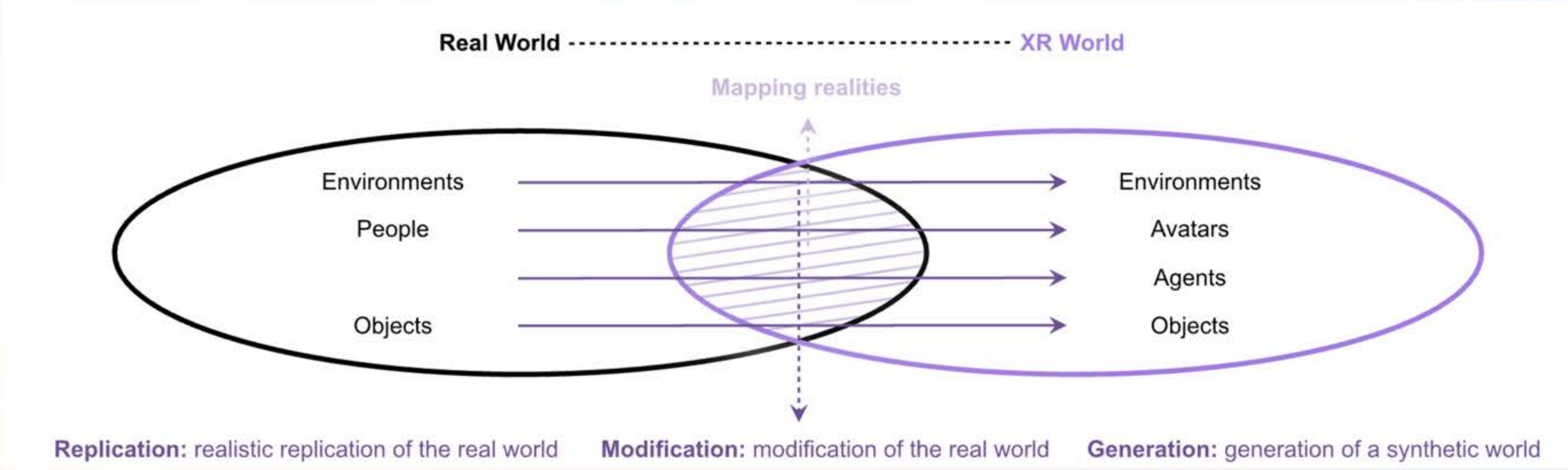
Session Agenda

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

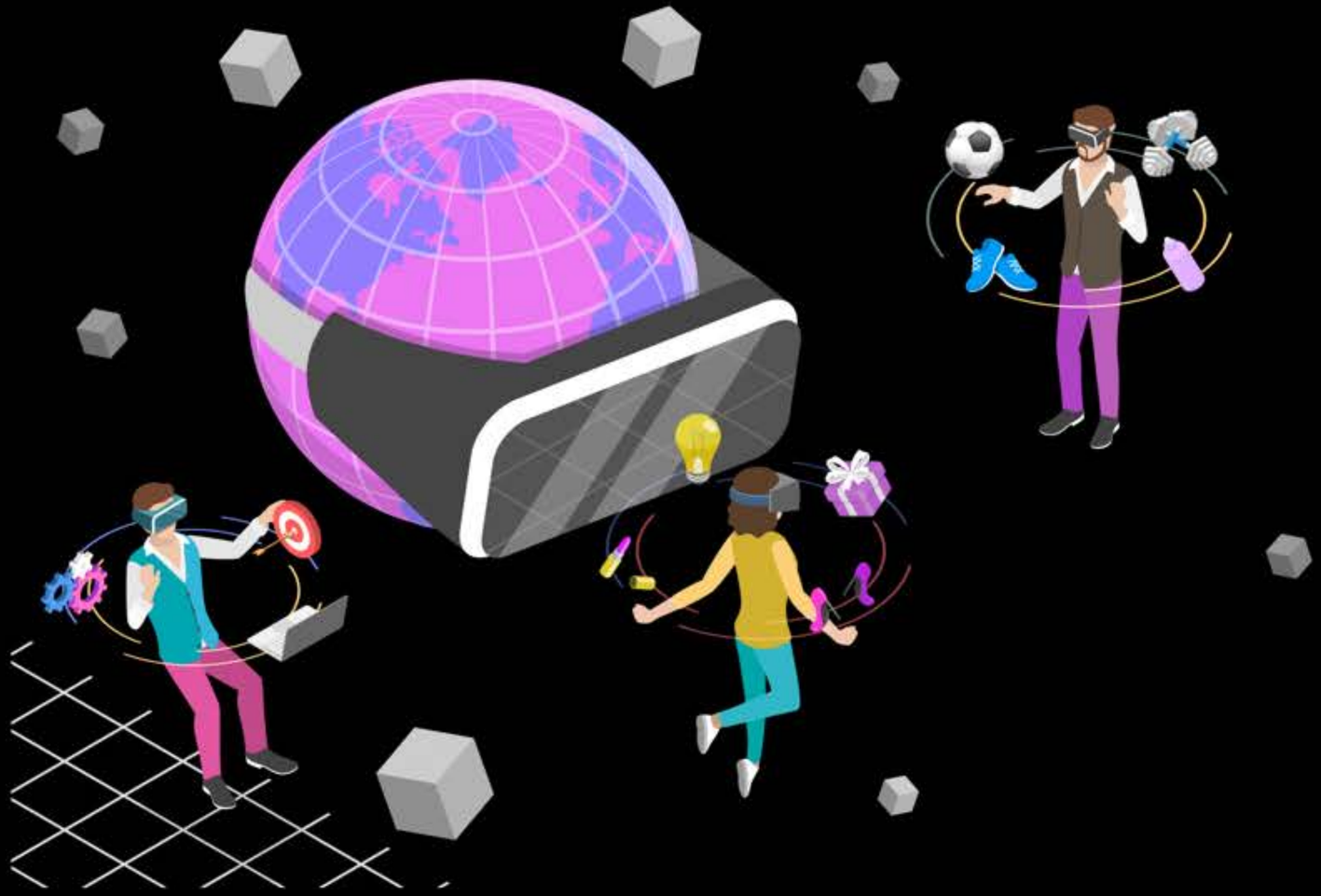
(2) Using AI to understand users in XR

(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



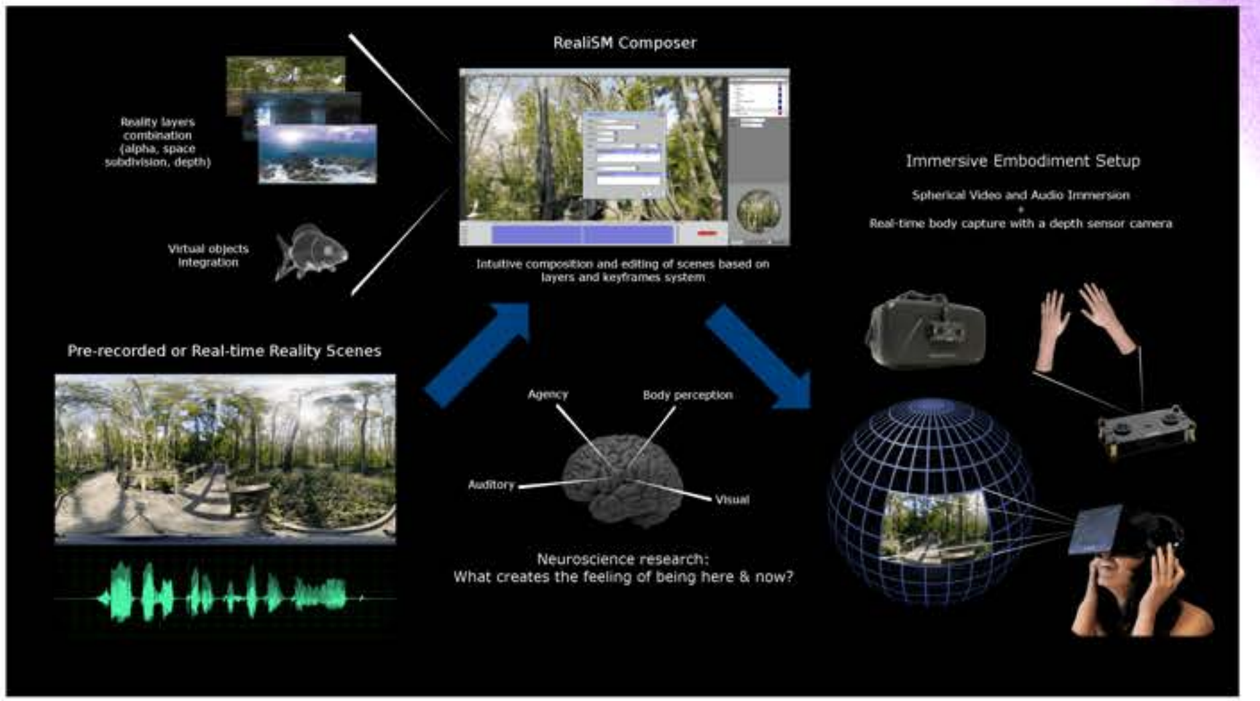
Using AI to Create XR Worlds

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.



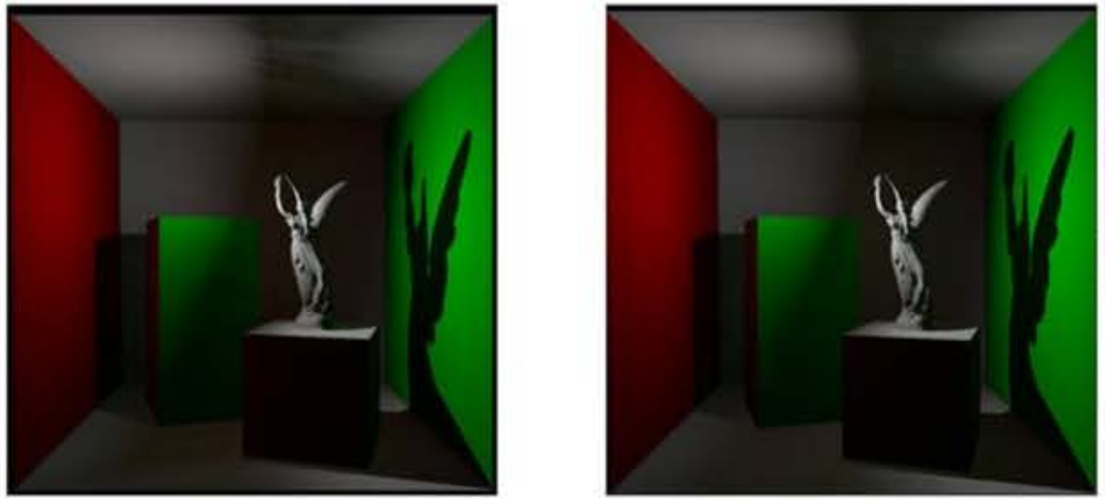
Creating XR Environments with AI

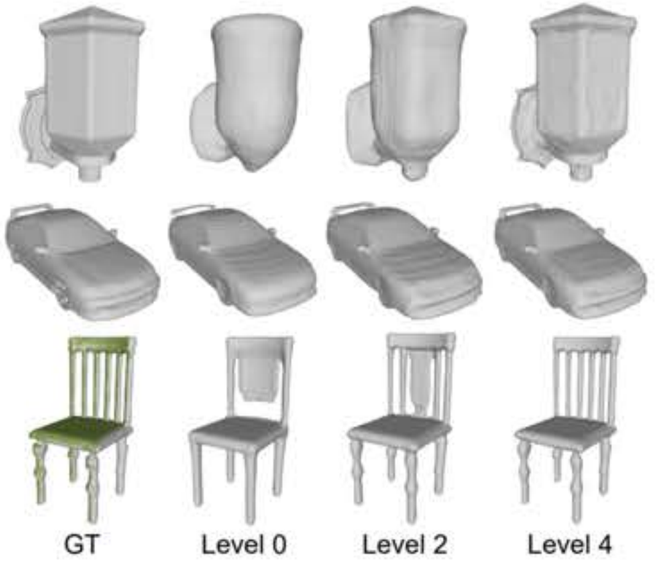
3. Measuring and Optimizing Illumination: AI models for dynamic lighting and shadow adjustments.

- Example: CNNs for realistic lighting conditions.

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

- Example: Deep learning for enhancing image details.





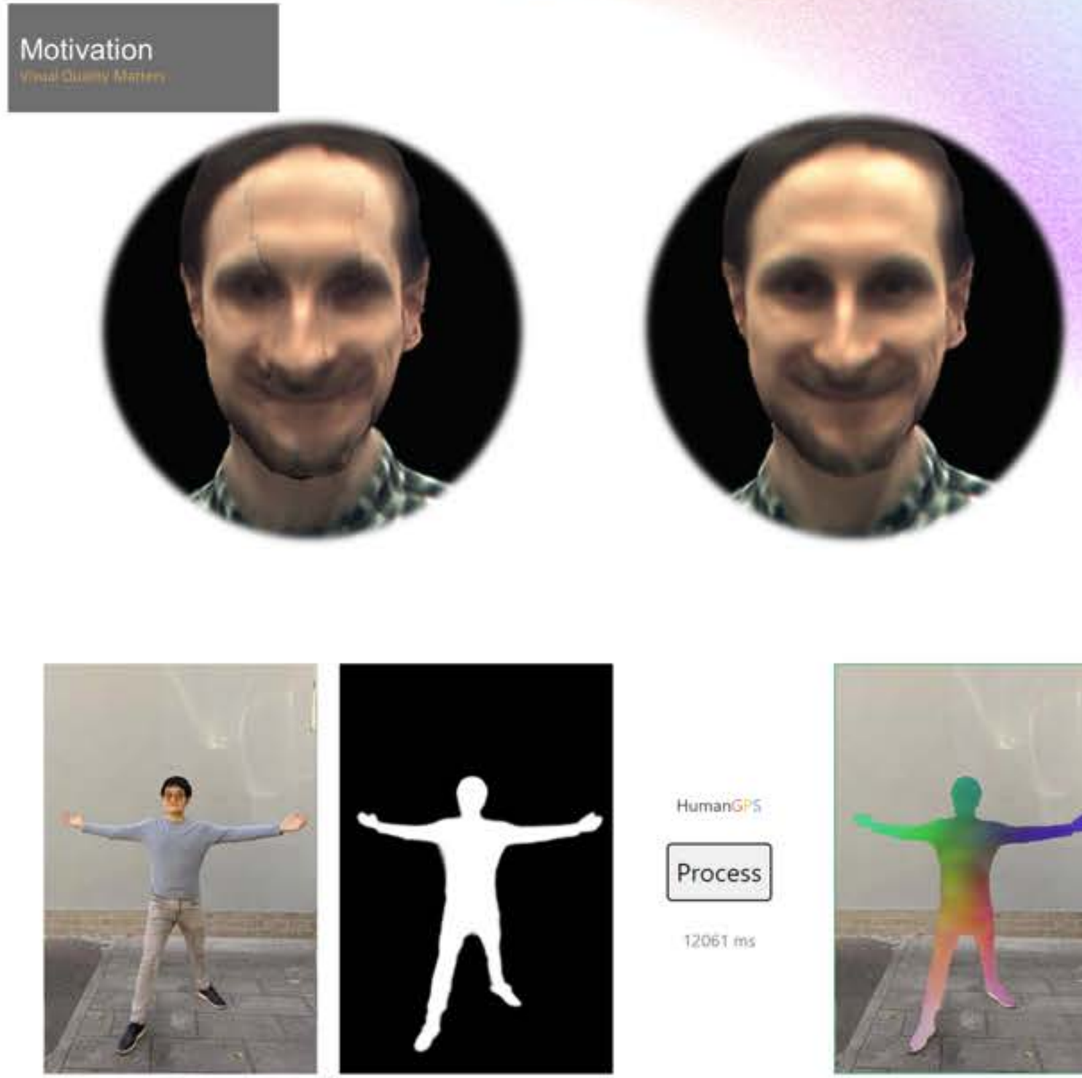
Creating XR Environments with AI

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.



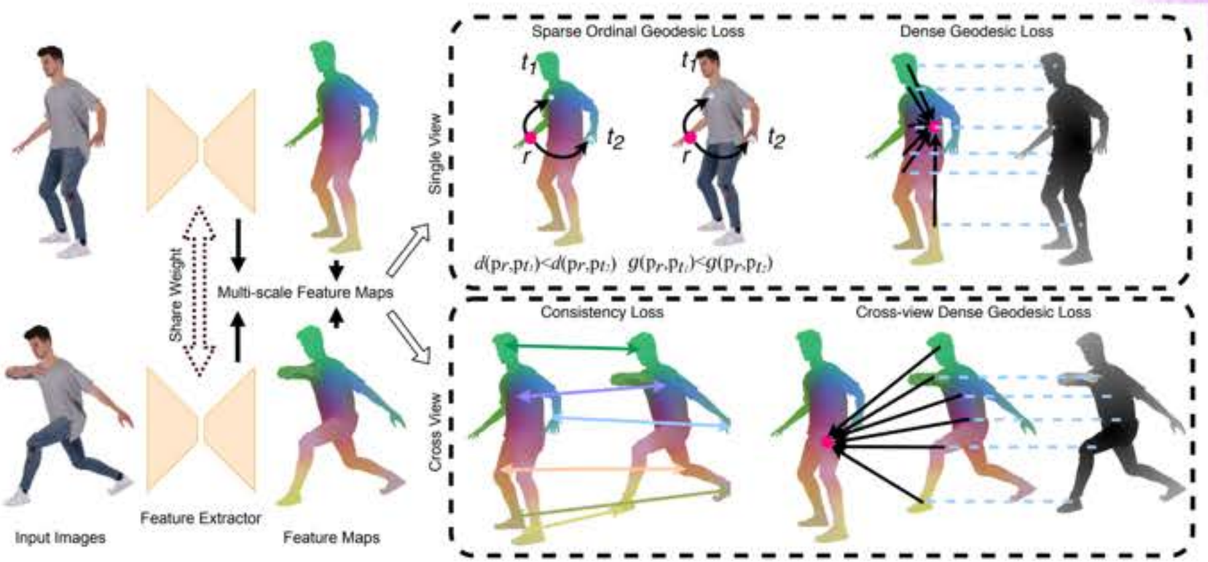
Creating Avatars

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.

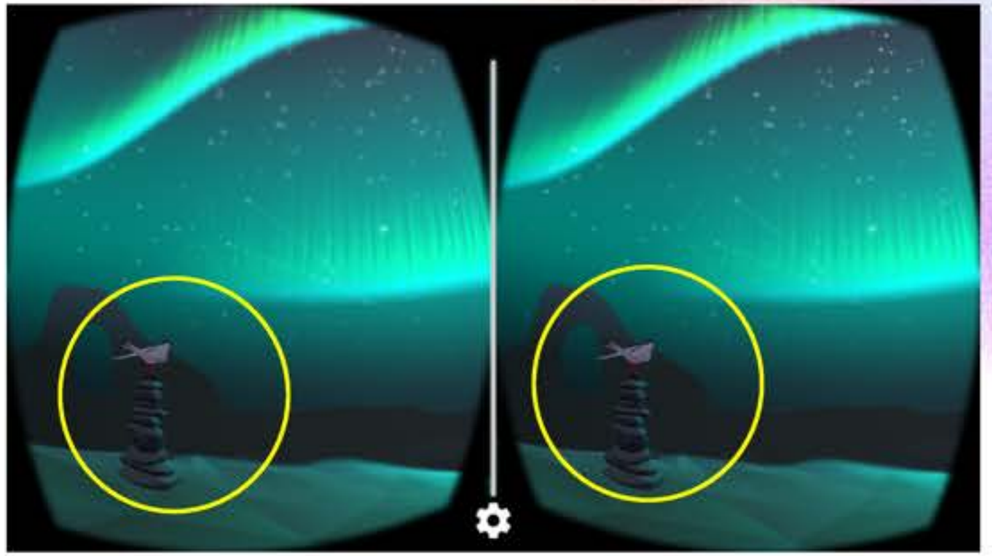


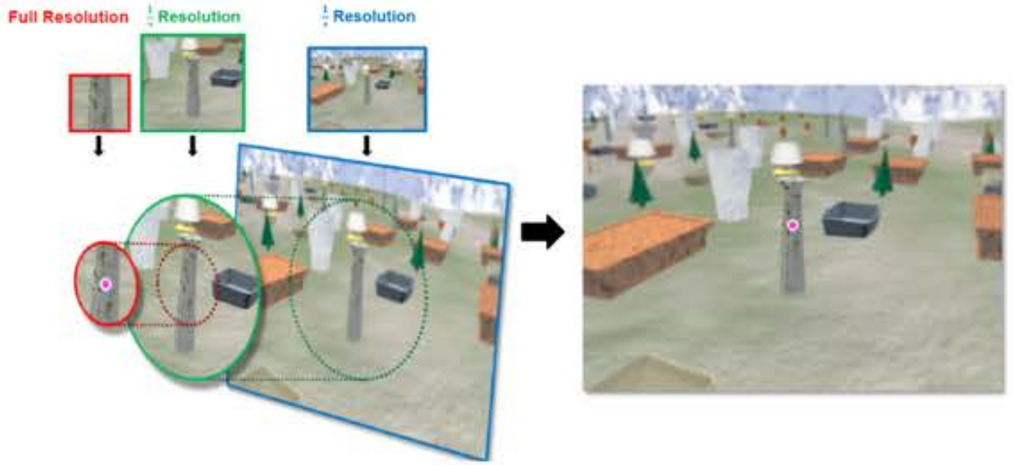
Creating Agents

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

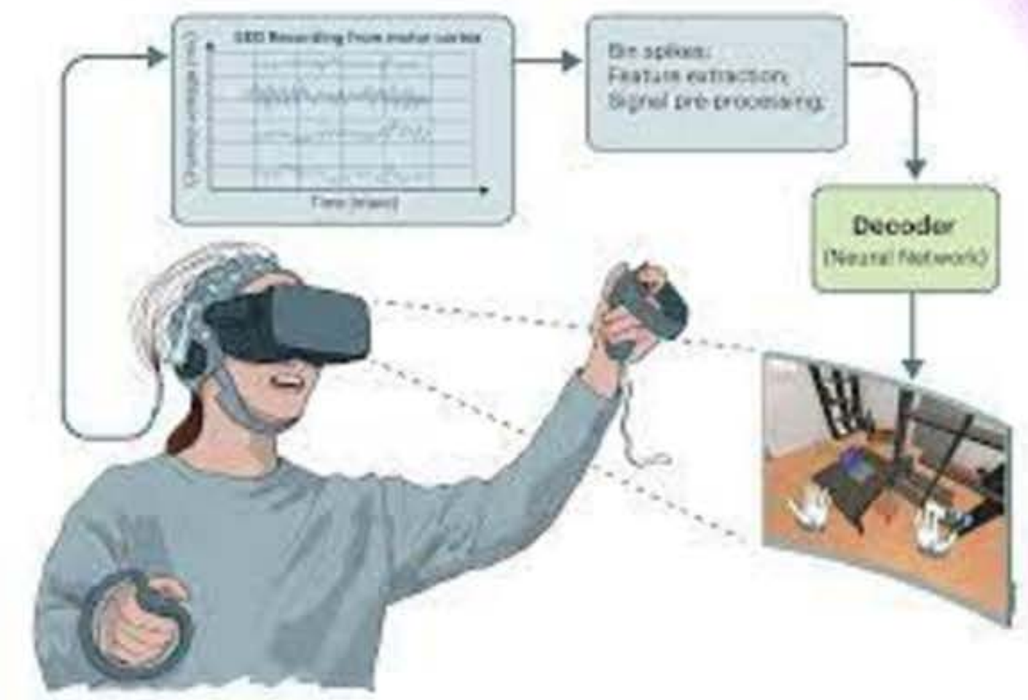
Predicting VR Sickness

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.

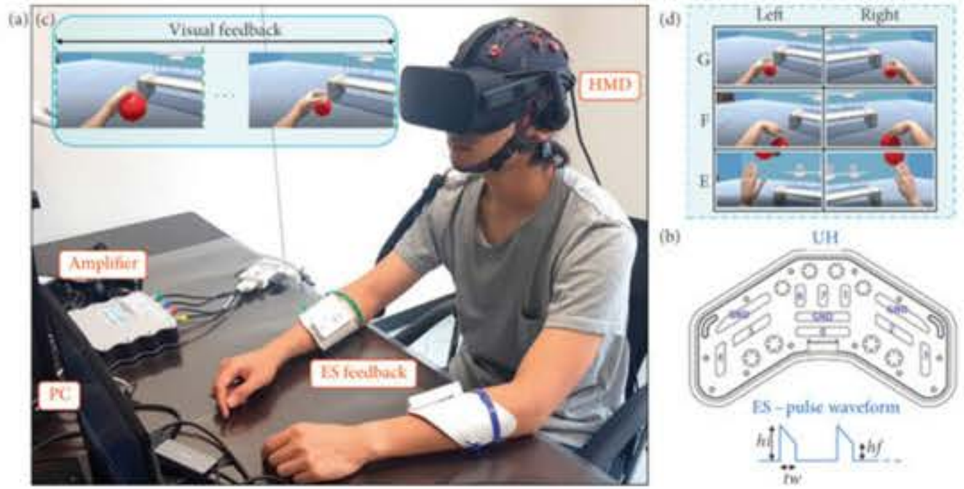


Predicting User Characteristics

1. **User Profiling**: AI techniques for real-time user profiling based on behavior and interactions.
    - Example: Clustering and classification methods for user segmentation.

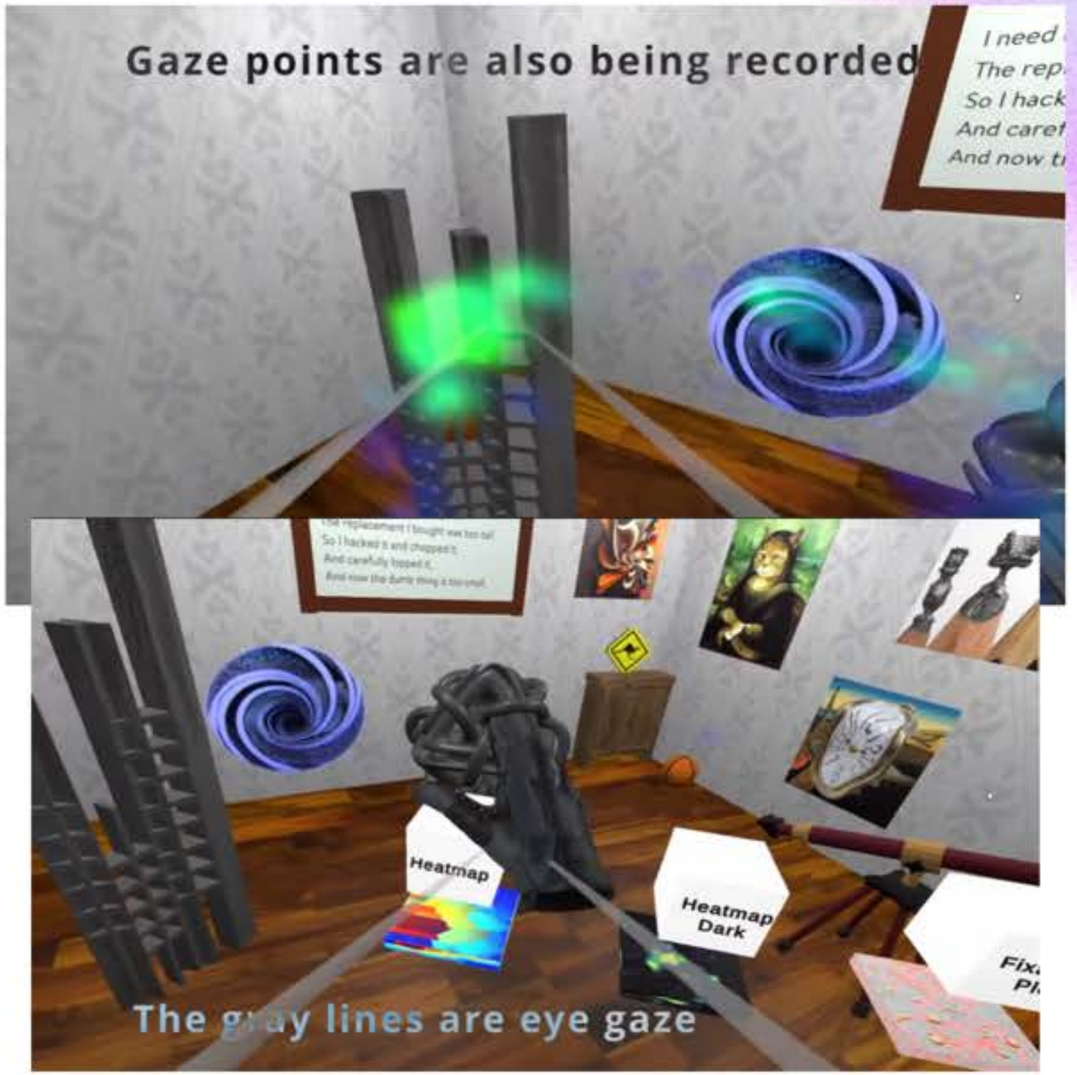
2. Predicting Viewport and Head Movement: AI models for predicting user head movements and viewport changes.

- Example: Reinforcement learning for optimal viewport prediction.

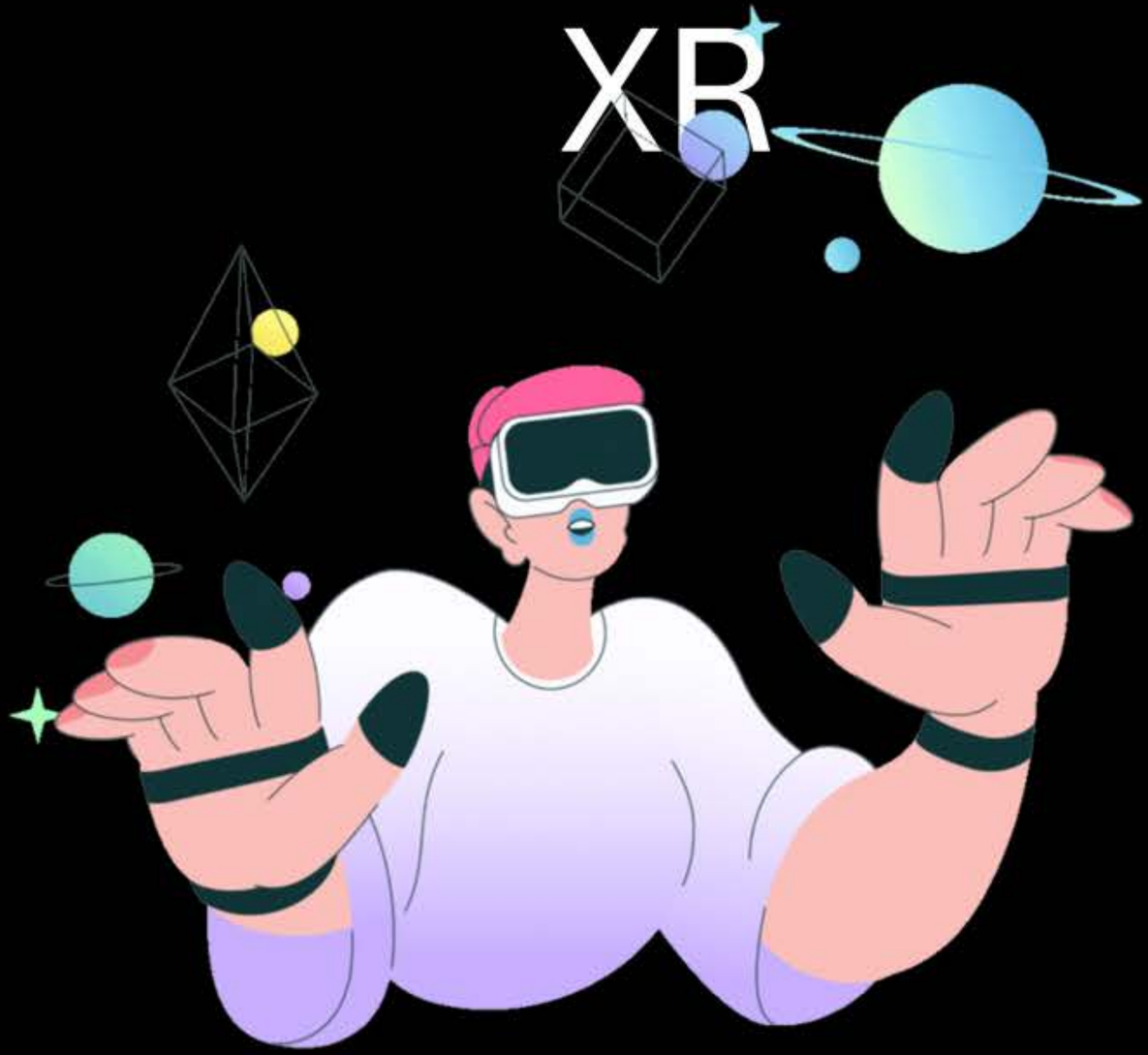


Eye Tracking and Gaze Analysis

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 XR

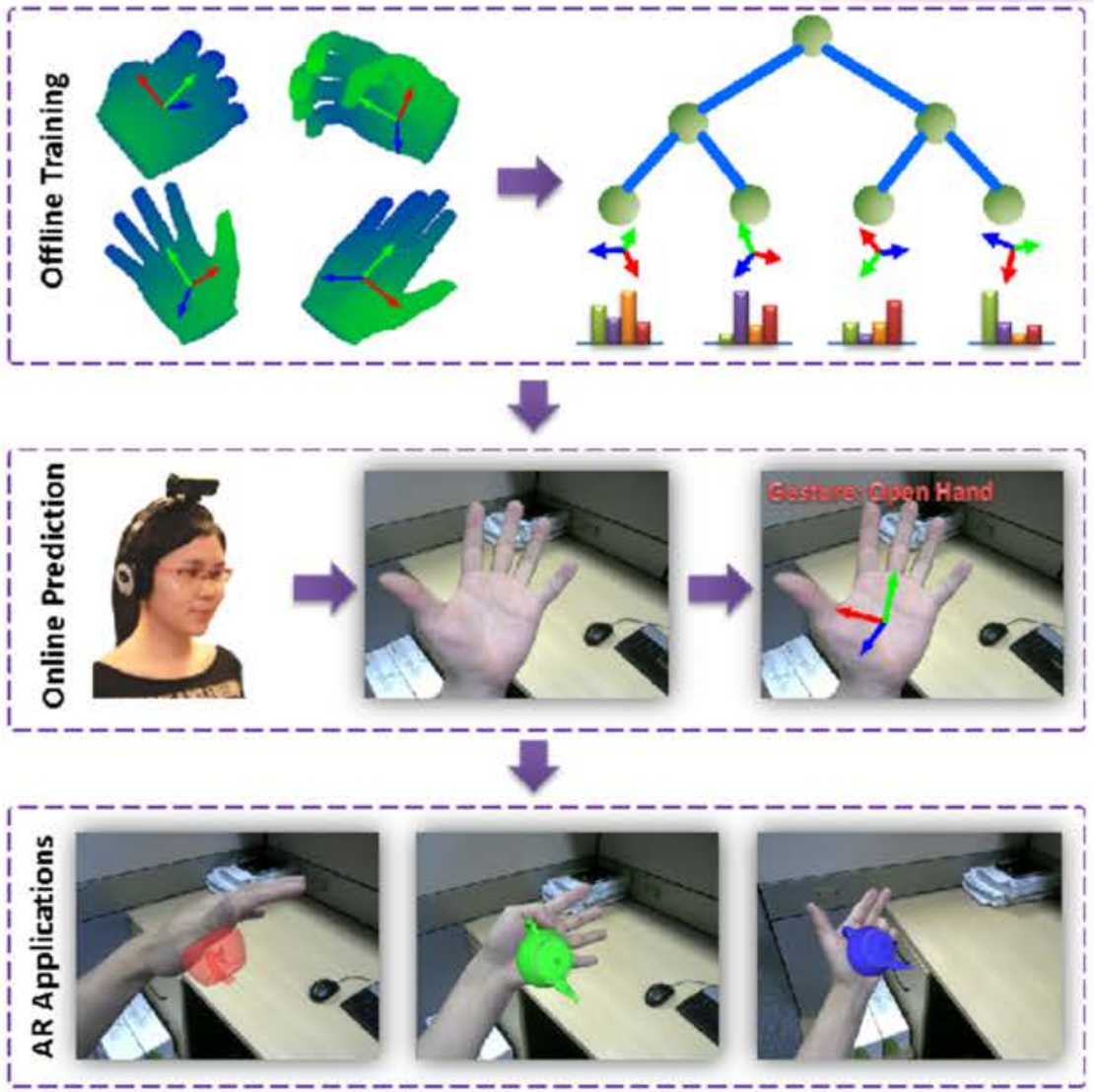


Using AI to Support Interaction

1. Gesture-based Interaction
2. Locomotion Techniques
3. Novel Devices and Interaction Techniques
4. Haptic Feedback

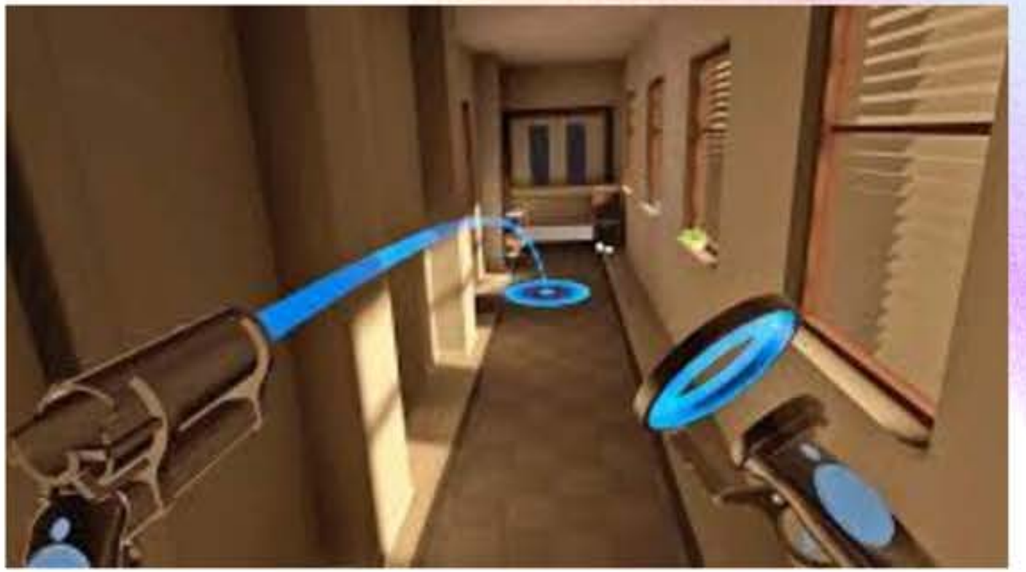
Gesture-Based Interaction

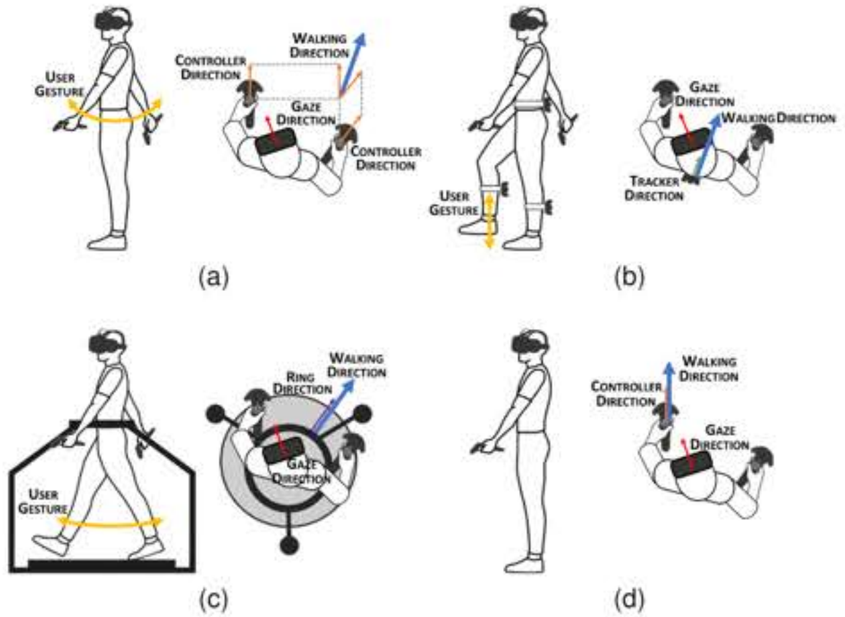
1. **3D Mid-Air Gesture Interaction**: AI methods for recognizing and responding to mid-air gestures.
   - Example: CNNs for hand gesture recognition.
2. **Gesture Recognition and Classification**: AI models for classifying complex gestures in real-time.
   - Example: LSTM networks for dynamic gesture classification.



Locomotion Techniques

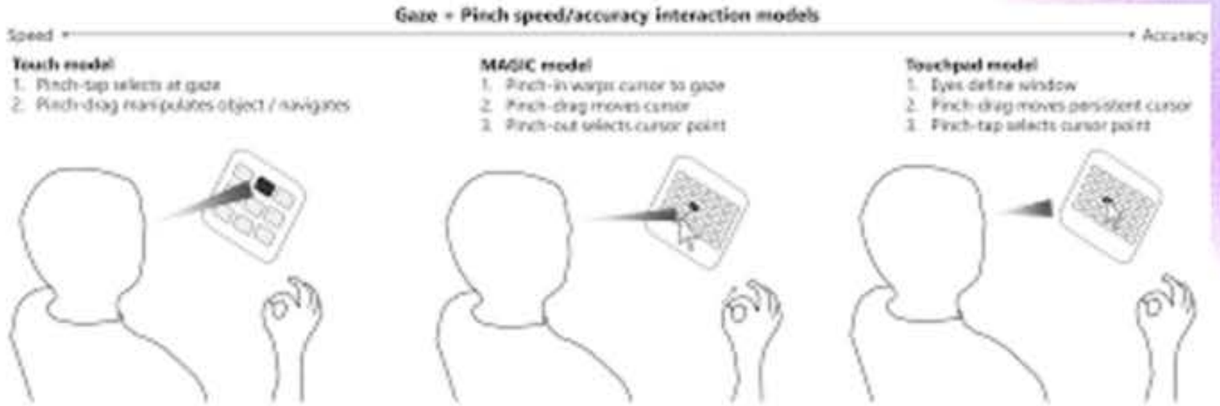
1. **Redirected Walking Techniques**: AI models for seamless navigation in confined VR spaces.
    - Example: Reinforcement learning for optimizing redirected walking paths.
2. **General Locomotion Techniques**: AI-enhanced techniques for natural and intuitive locomotion.
    - Example: Predictive models for dynamic path adjustment.





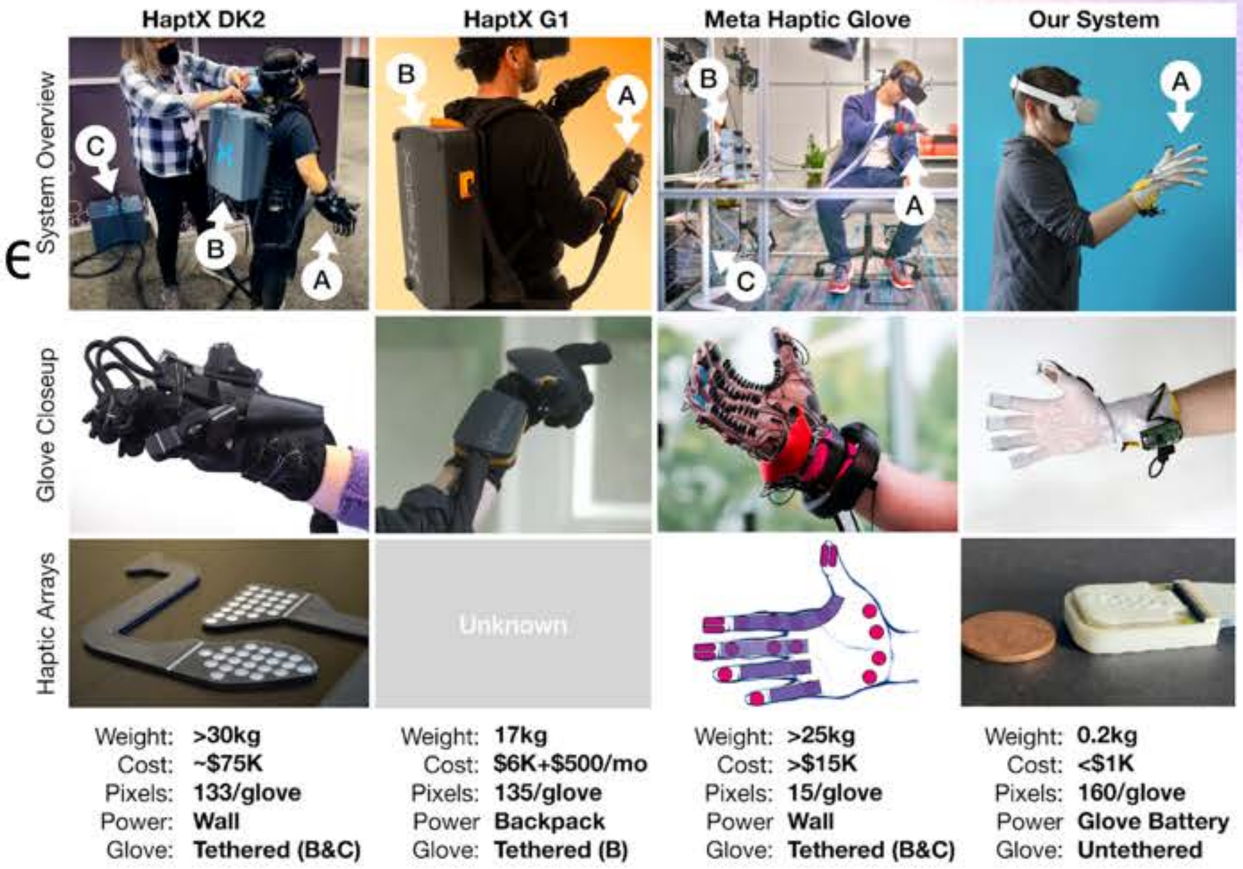
Novel Devices and Interaction Techniques

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

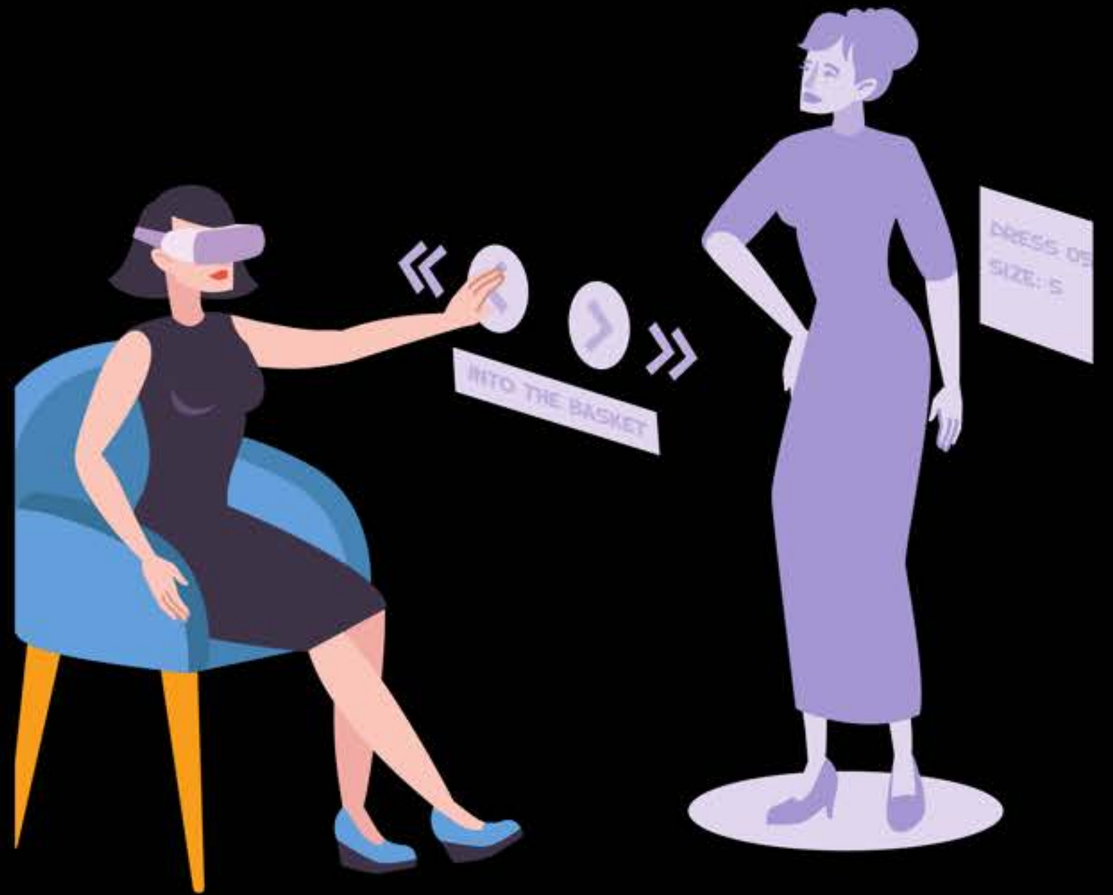


Haptic Feedback

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



Interaction with Intelligent Virtual agents



Interaction with Intelligent Virtual Agents

1

Interacting with Agents

2

Physical Interaction with Agents

3

Trust and Empathy in Agents

Interacting with Crowds of Agents

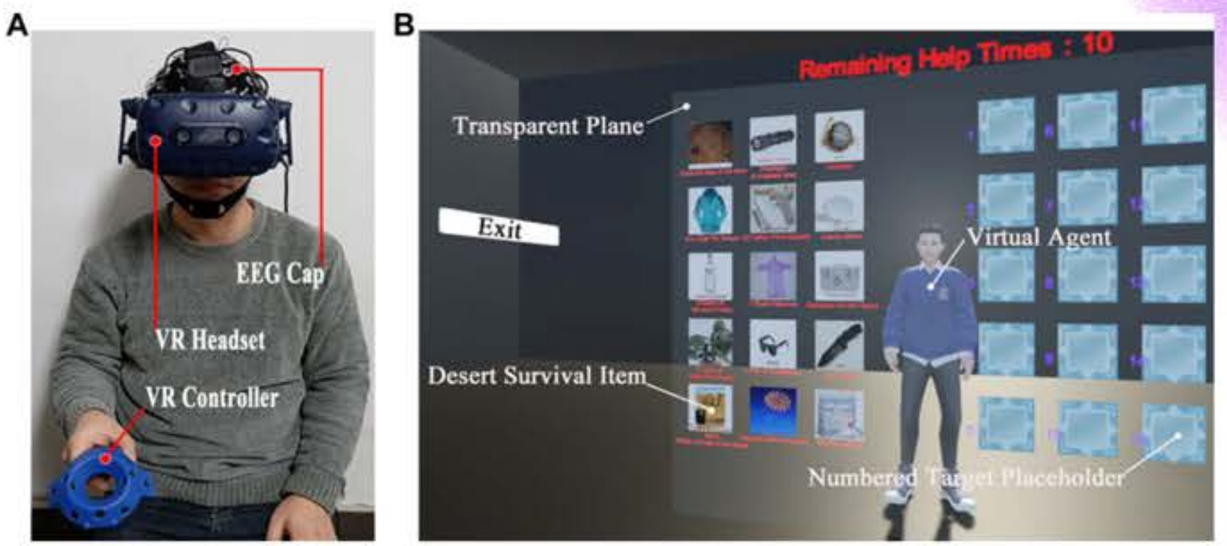
- **Crowd Behavior Modeling**: AI models for realistic crowd behaviors and dynamics.
  - Example: Multi-agent reinforcement learning for crowd simulation.

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

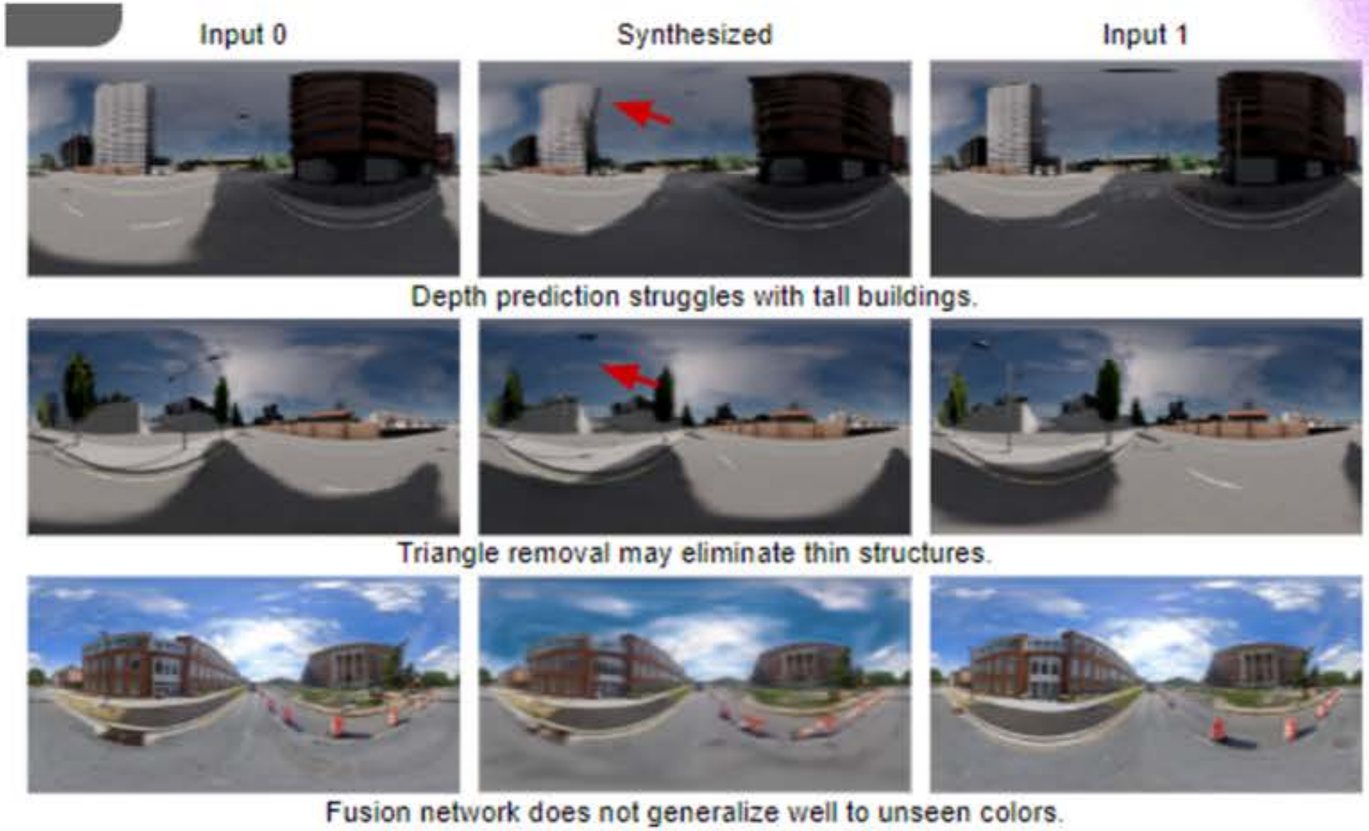
Using XR to Support AI Research

Visualising AI Methods in XR

Generating Training Data for XR

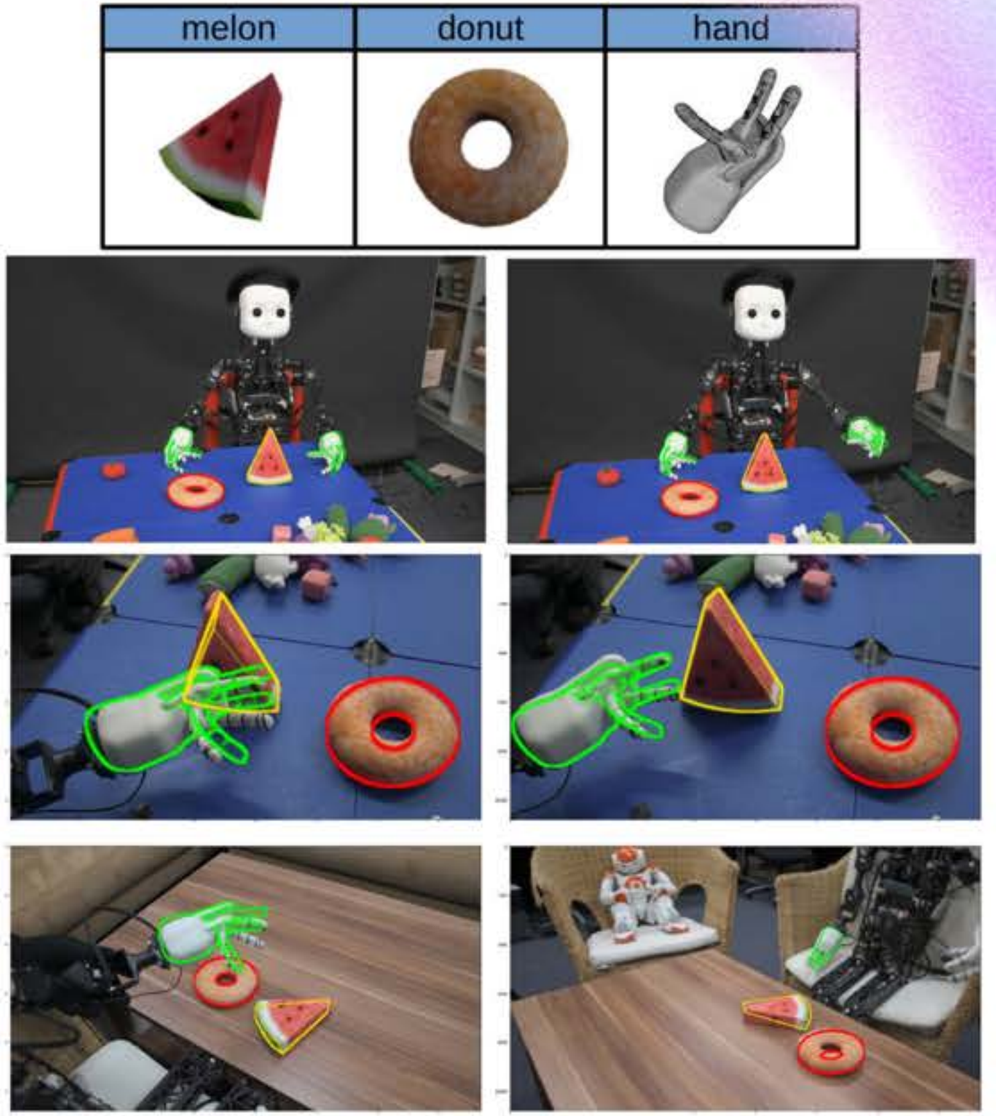
Visualizing AI Methods in XR

- **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

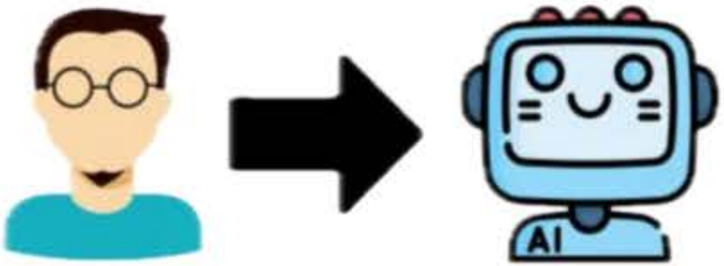
ADAPT WITH FEEDBACK

Adapt AI with user feedback

HELPFUL AI

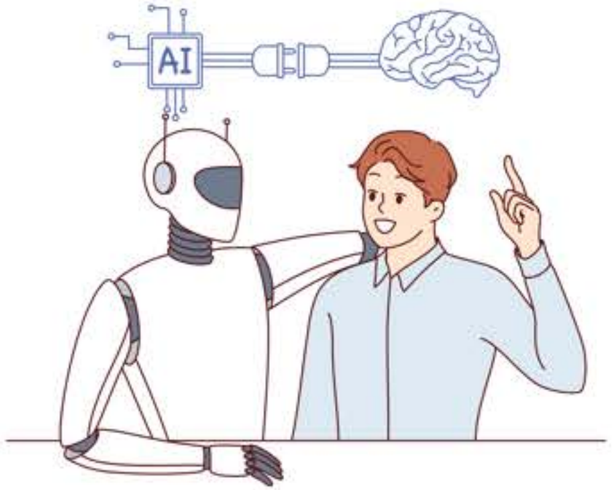
Create helpful AI that enhances work and play

Automation



Replace Human Intelligence with Artificial Intelligence

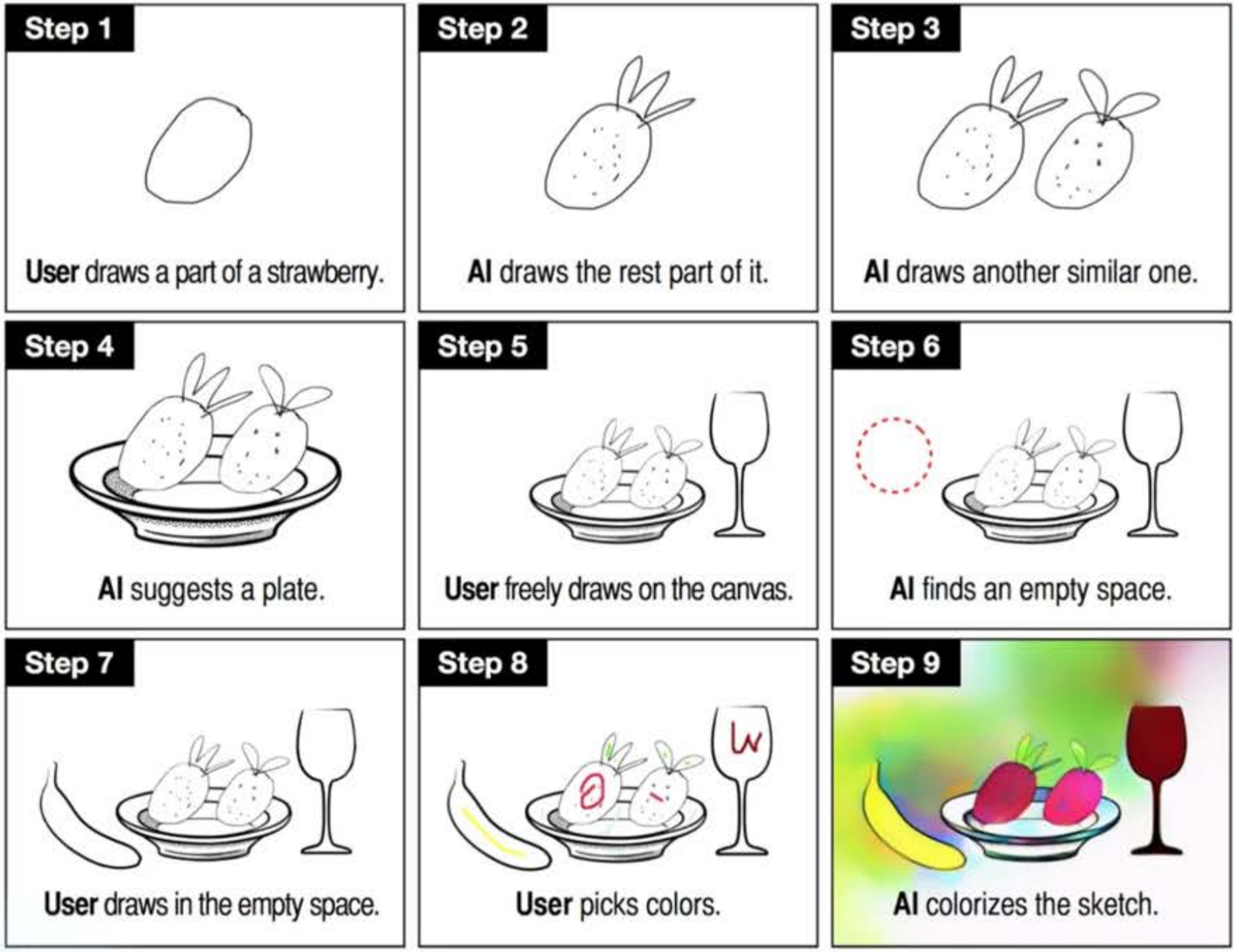
Collaboration



Human Intelligence with Artificial Intelligence

Augmentation

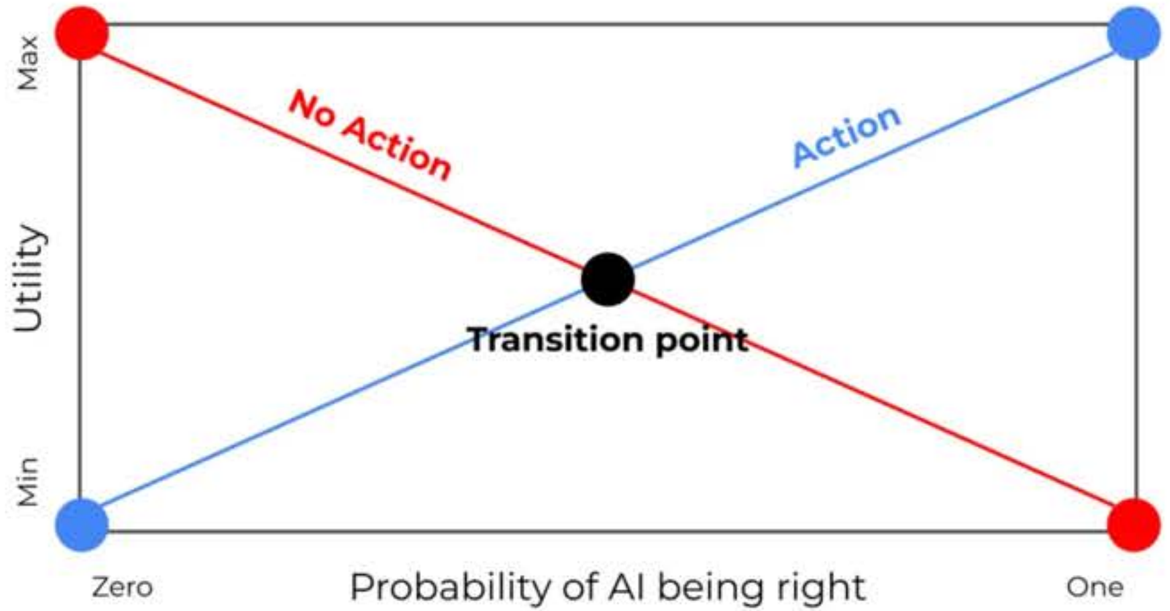
Augment Human Intelligence with Artificial Intelligence



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 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 and creating design frameworks

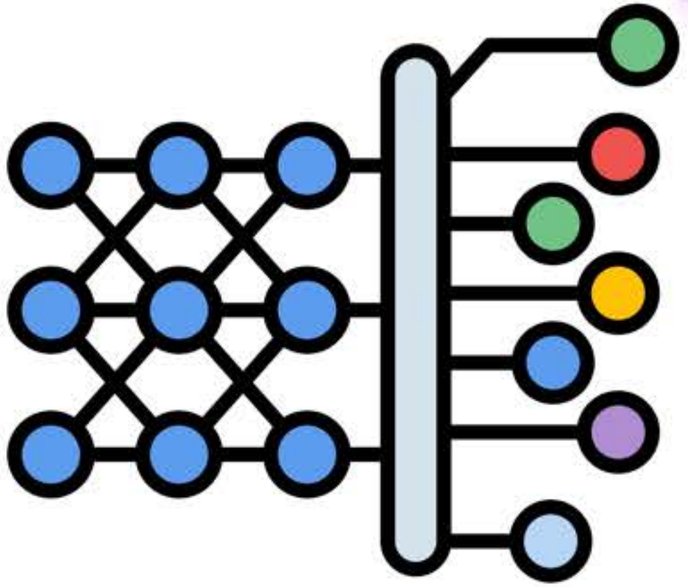
Tools and Frameworks for AI for XR

Programming - AI Tools

Machine and Deep Learning Frameworks

Use Python Programming Language

Follow Framework



Develop ML/DL Models Classification / Regression/ Generation

Generative AI Tools

OpenAI

Mistral AI

Midjourney

Gemini

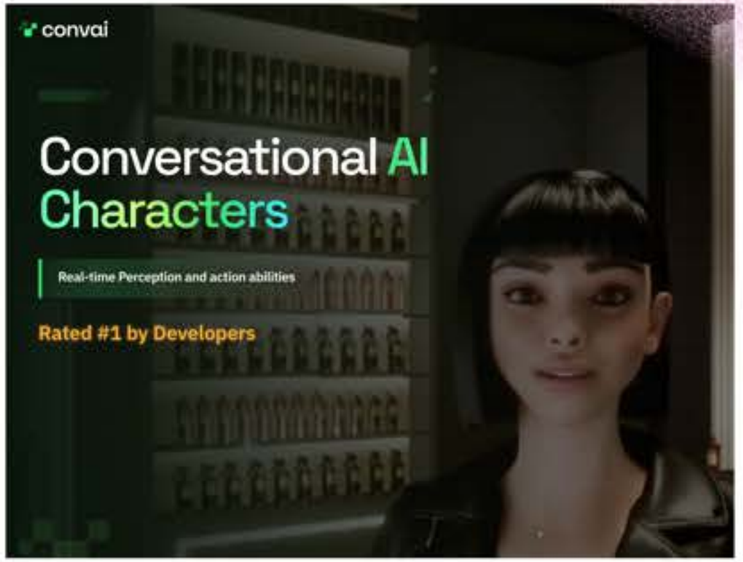
deepseek

Claude

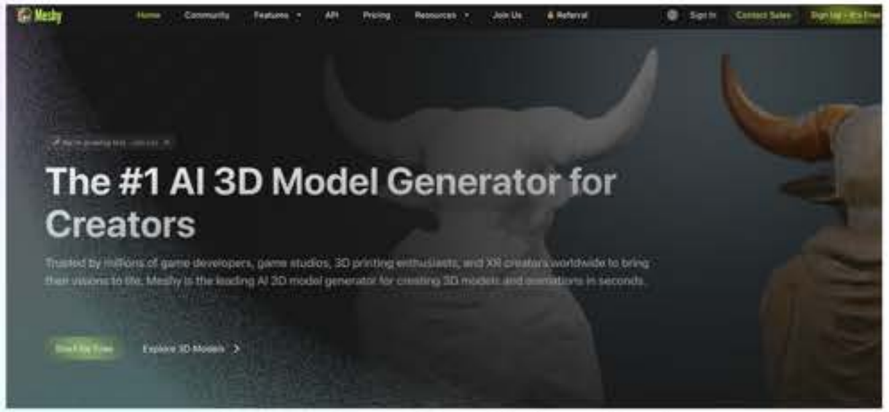
Generative AI Tools for XR

Blockade Labs

Text to SkyBox Generation



Conversational Agents

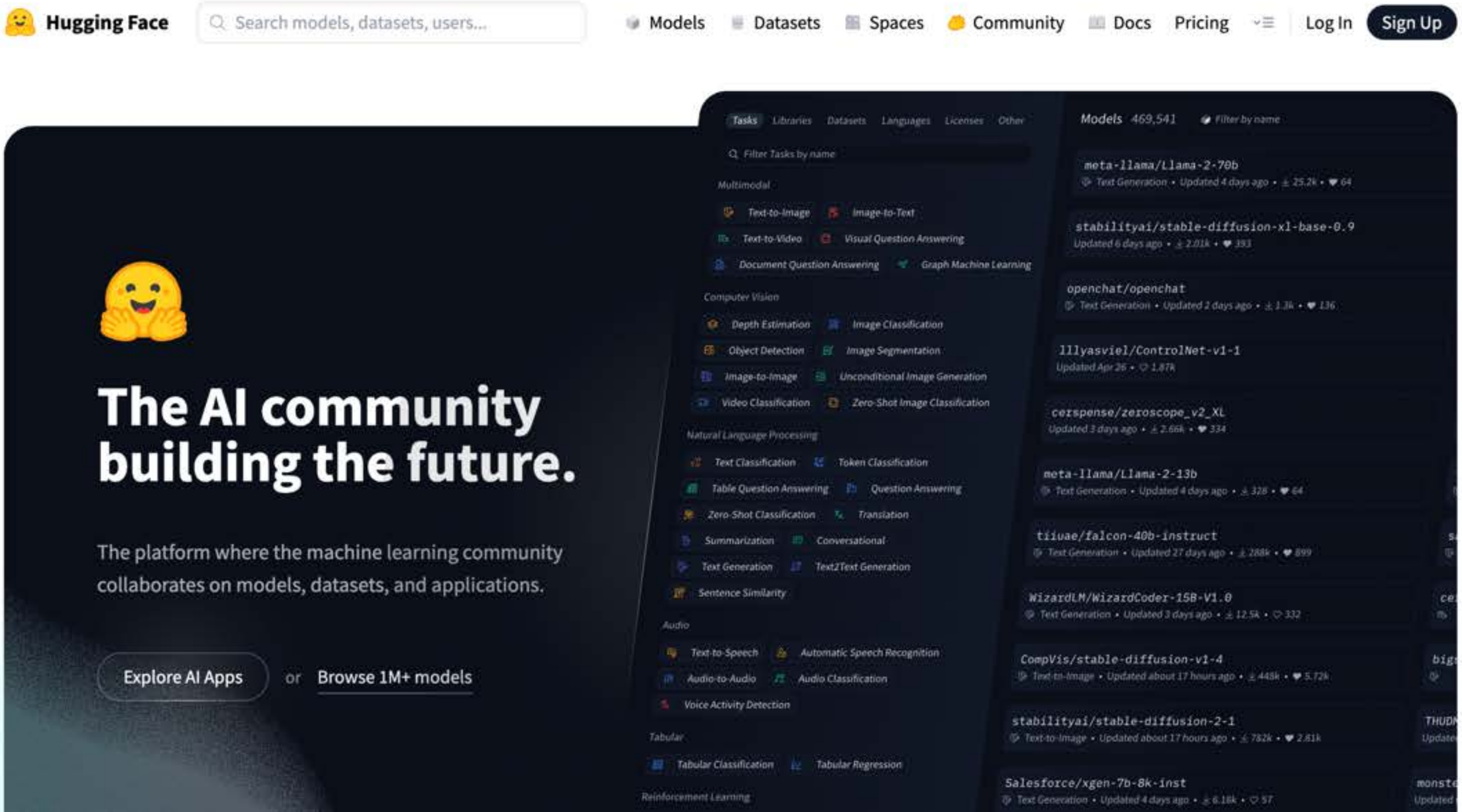


Text to 3D Models

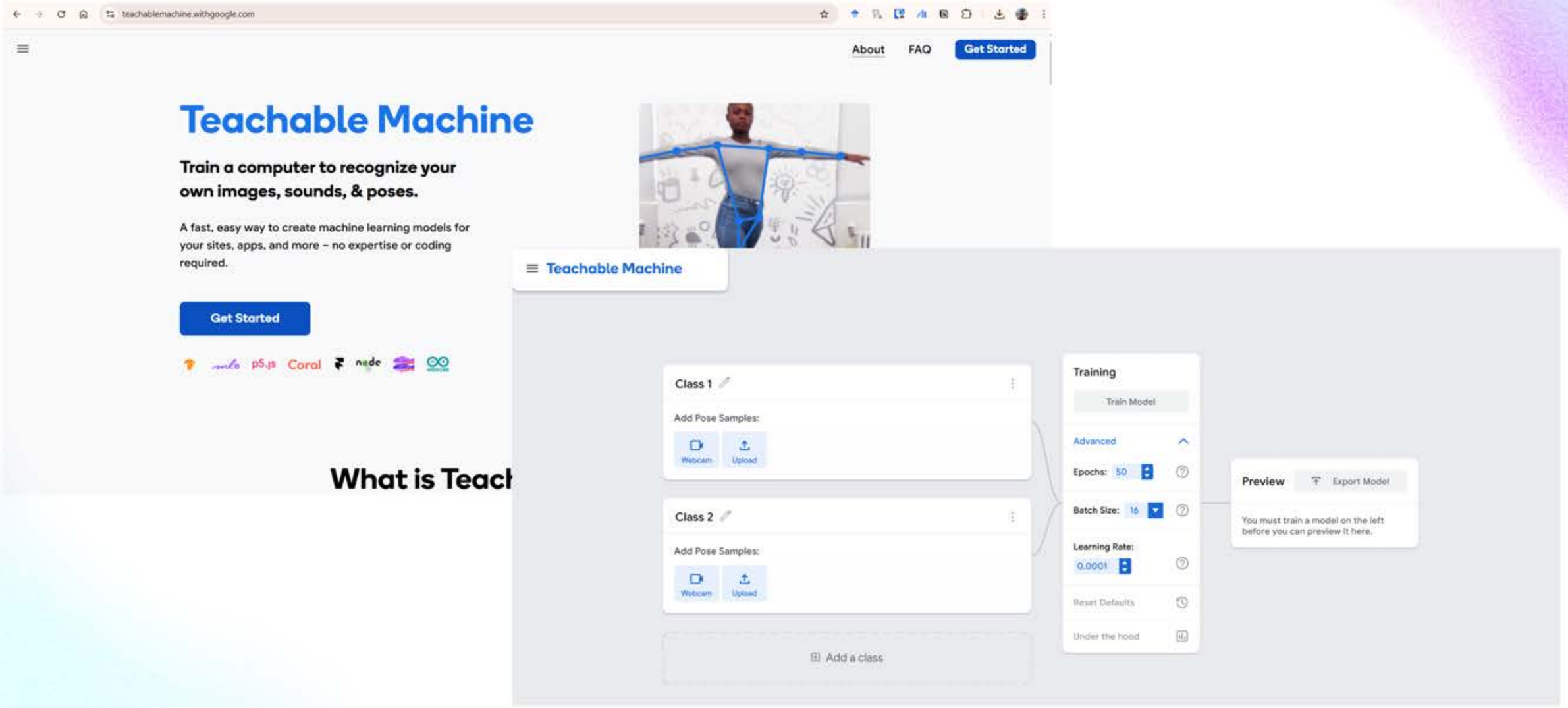
MusicGen

Text to Music

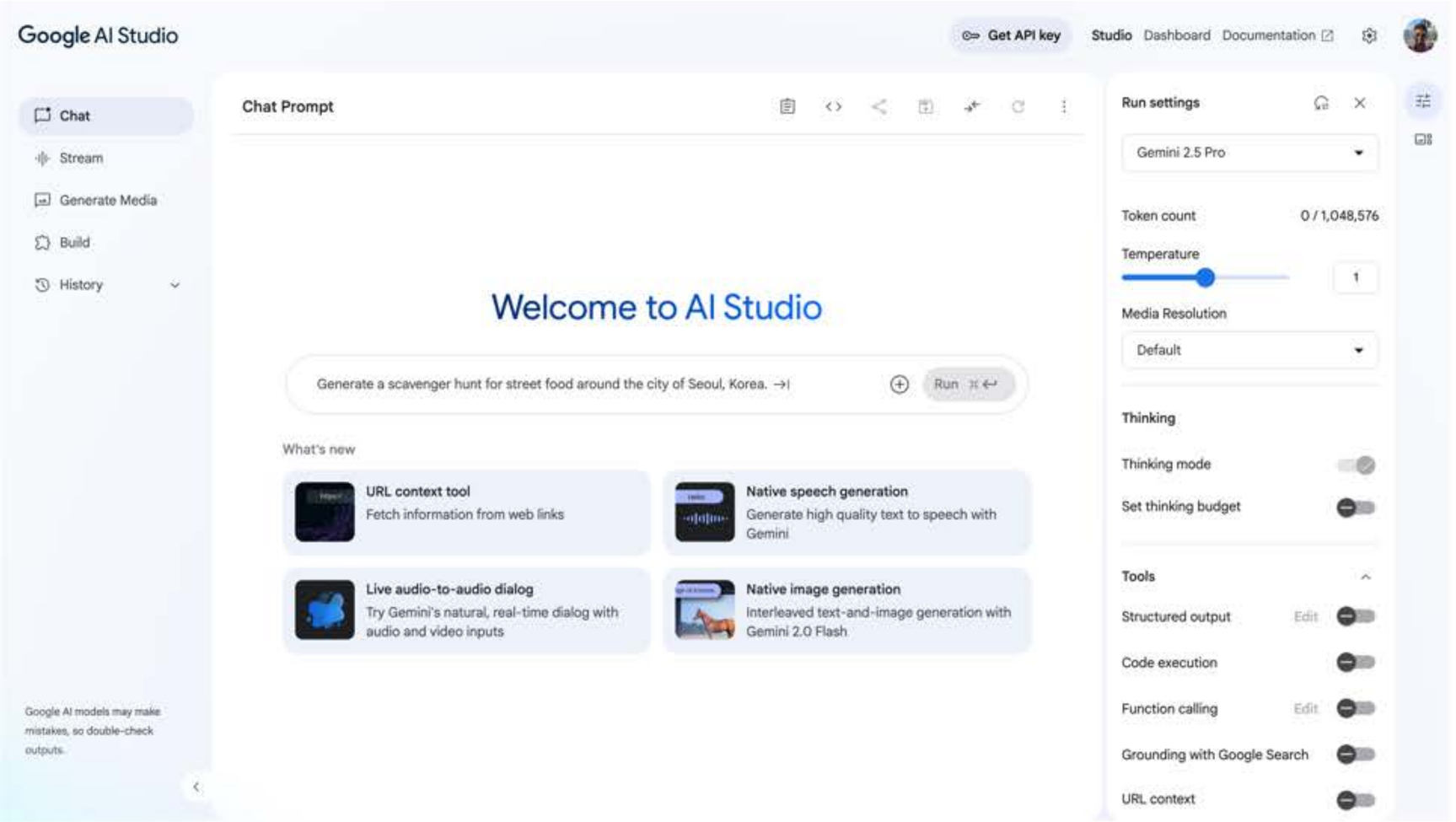
Finding AI Models for your Applications



Teachable Machine



Google AI Studio



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.

OLLAMA

- 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

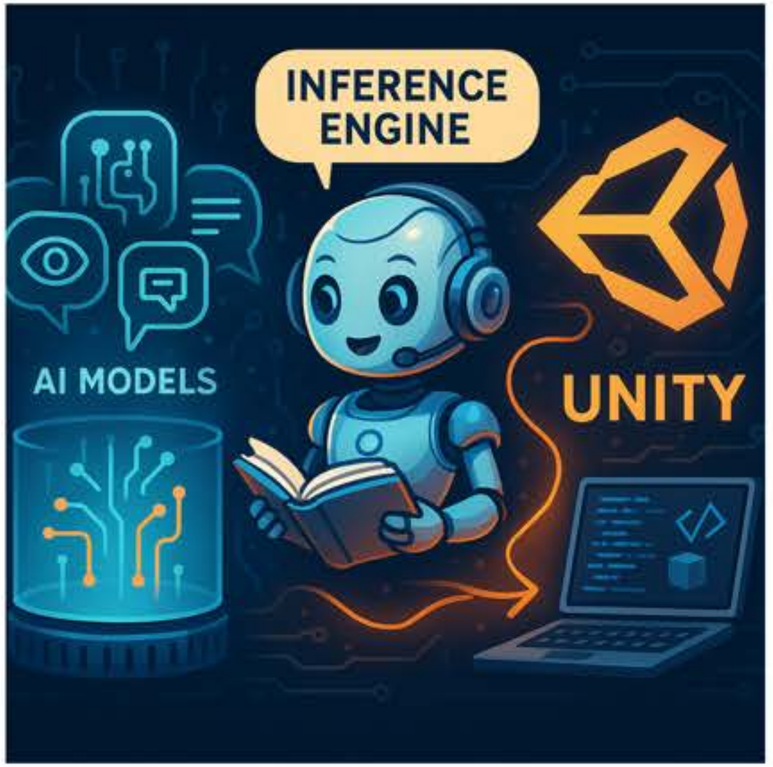
3. Inference Engine

The powerhouse

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

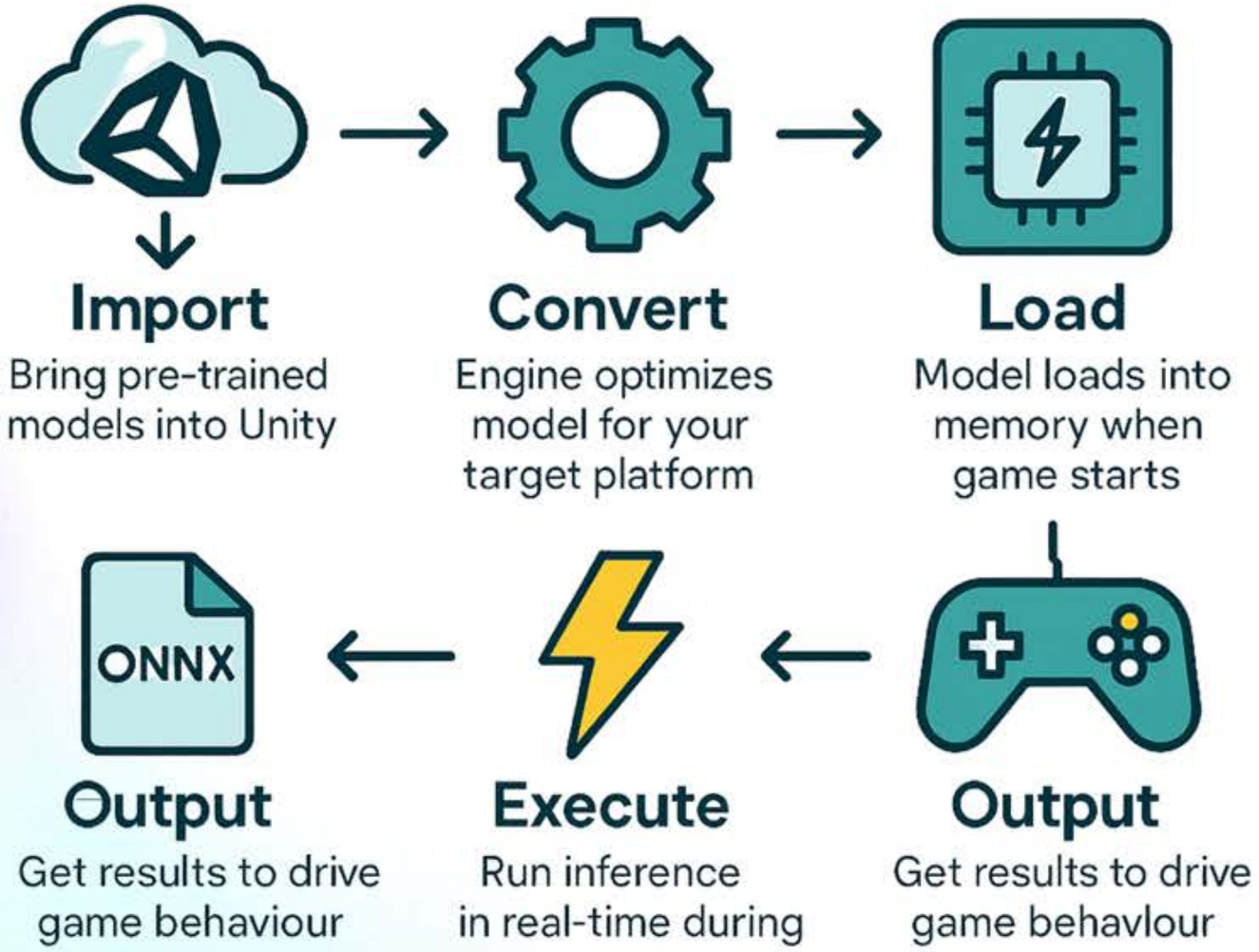
Unity AI Inference Engine

- 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.

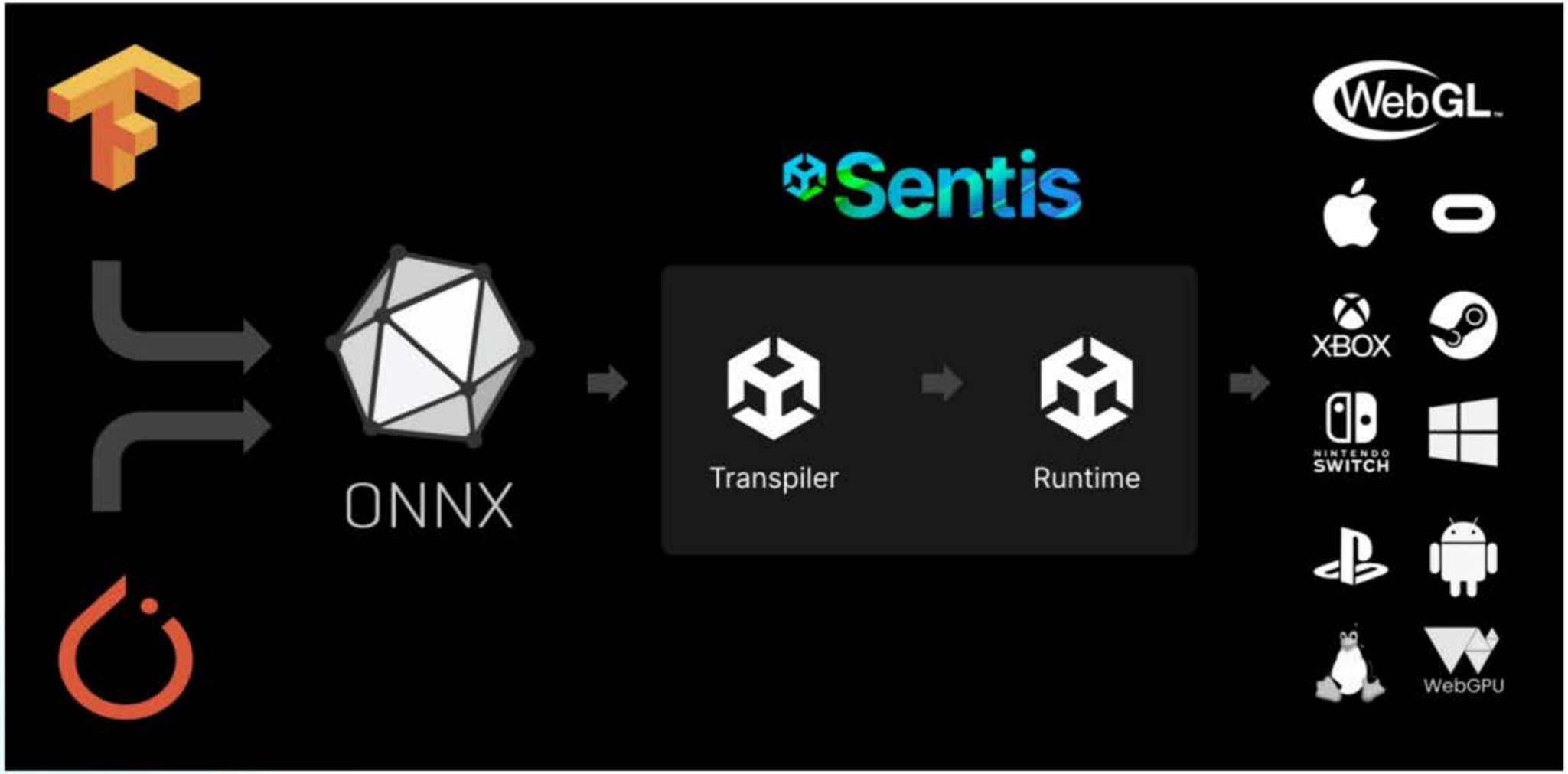
Model Support:

ONNX format (industry standard)

CPU and GPU execution

CPU and GPU execution

Integrating AI Models in Unity



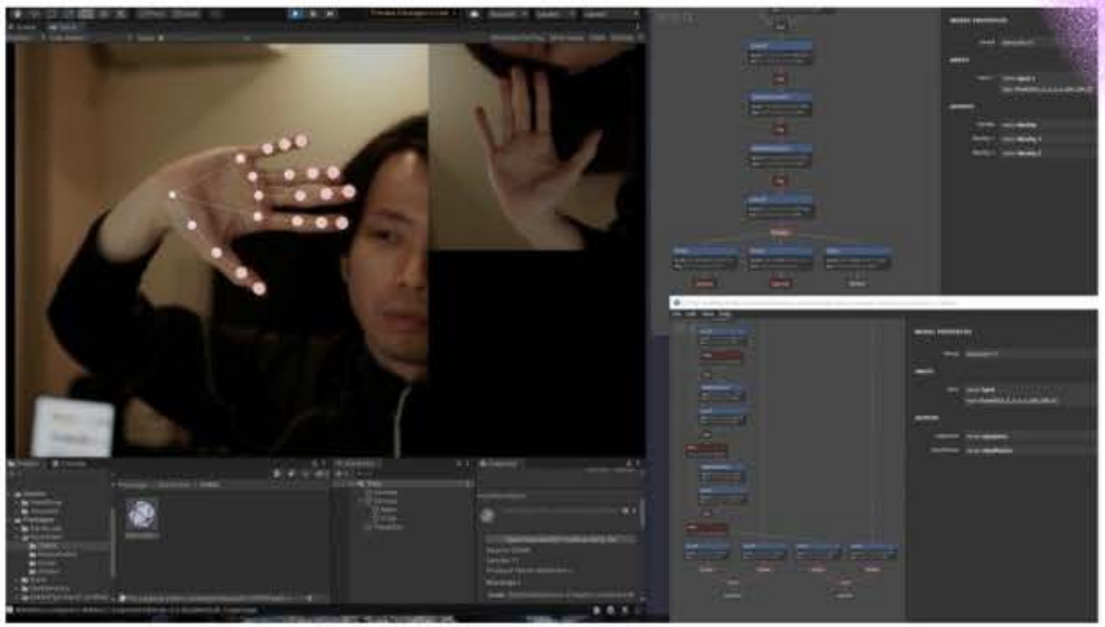
Setting Up Your First Model

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

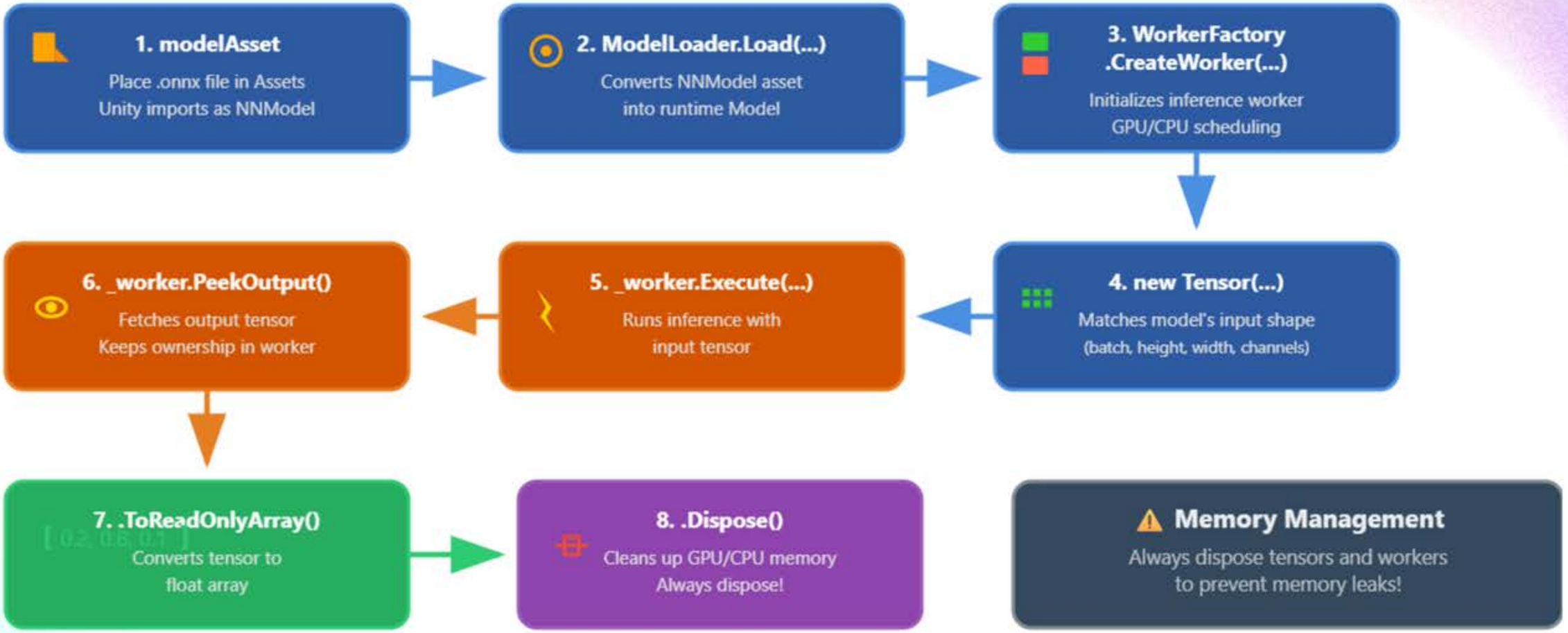
Step 4: Running Inference

```
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();
}
```


1. modelAsset
Place .onnx file in Assets
Unity imports as NNModel
2. ModelLoader.Load(...)
Converts NNModel asset
into runtime Model
3. WorkerFactory
.CreateWorker(...)
Initializes inference worker
GPU/CPU scheduling
4. new Tensor(...)
Matches model's input shape
(batch, height, width, channels)
5. _worker.Execute(...)
Runs inference with
input tensor
6. _worker.PeekOutput()
Fetches output tensor
Keeps ownership in worker
7. .ToReadOnlyArray()
Converts tensor to
float array
8. .Dispose()
Cleans up GPU/CPU memory
Always dispose!
Memory Management
Always dispose tensors and workers
to prevent memory leaks!

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

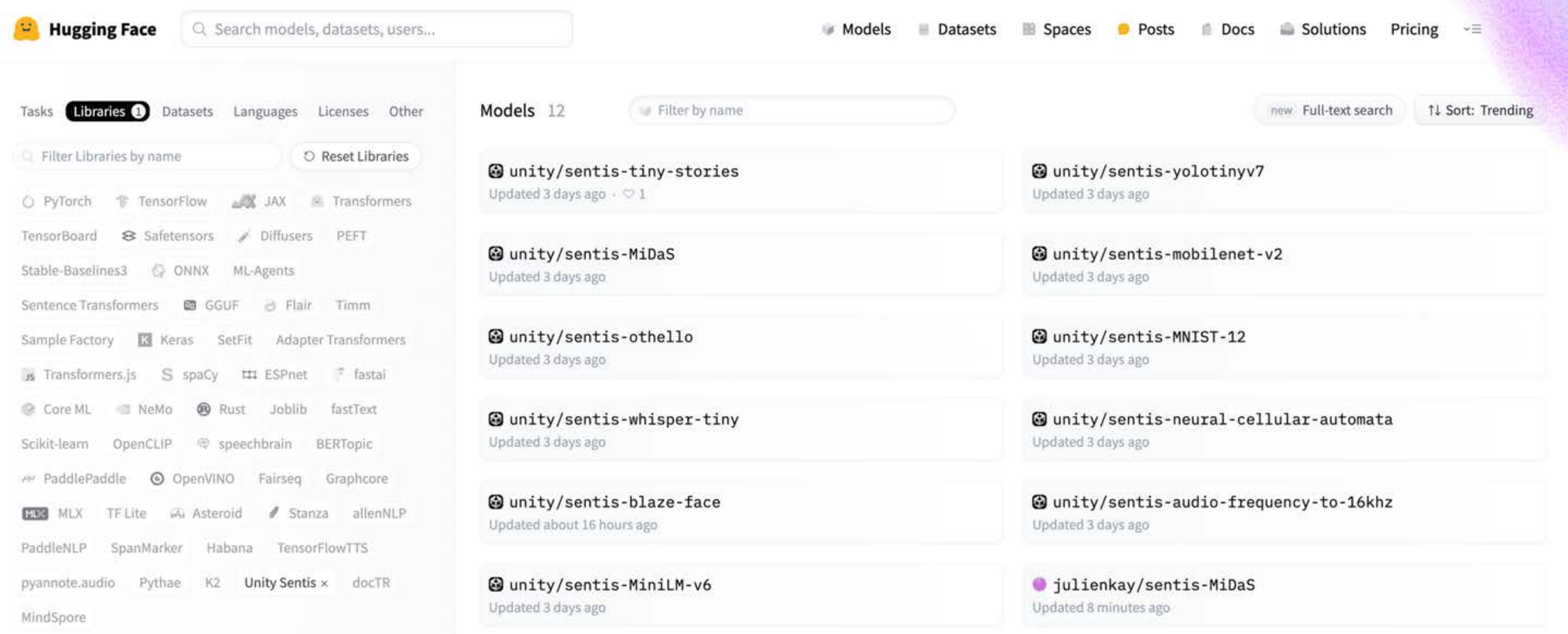
Async Execution

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



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

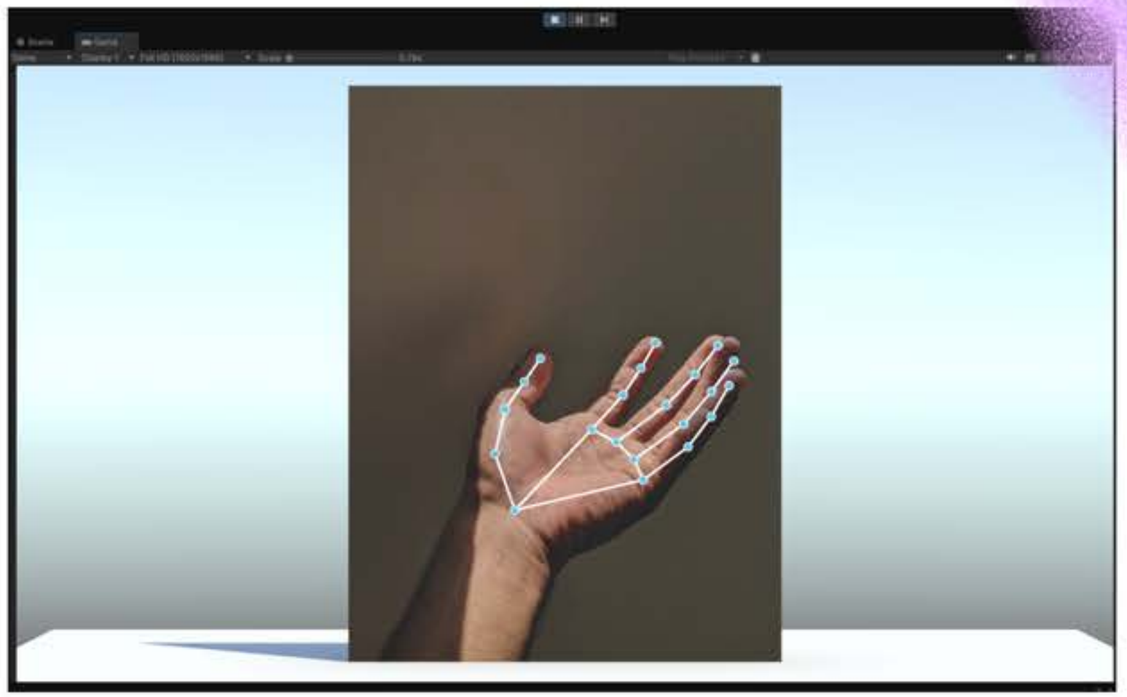
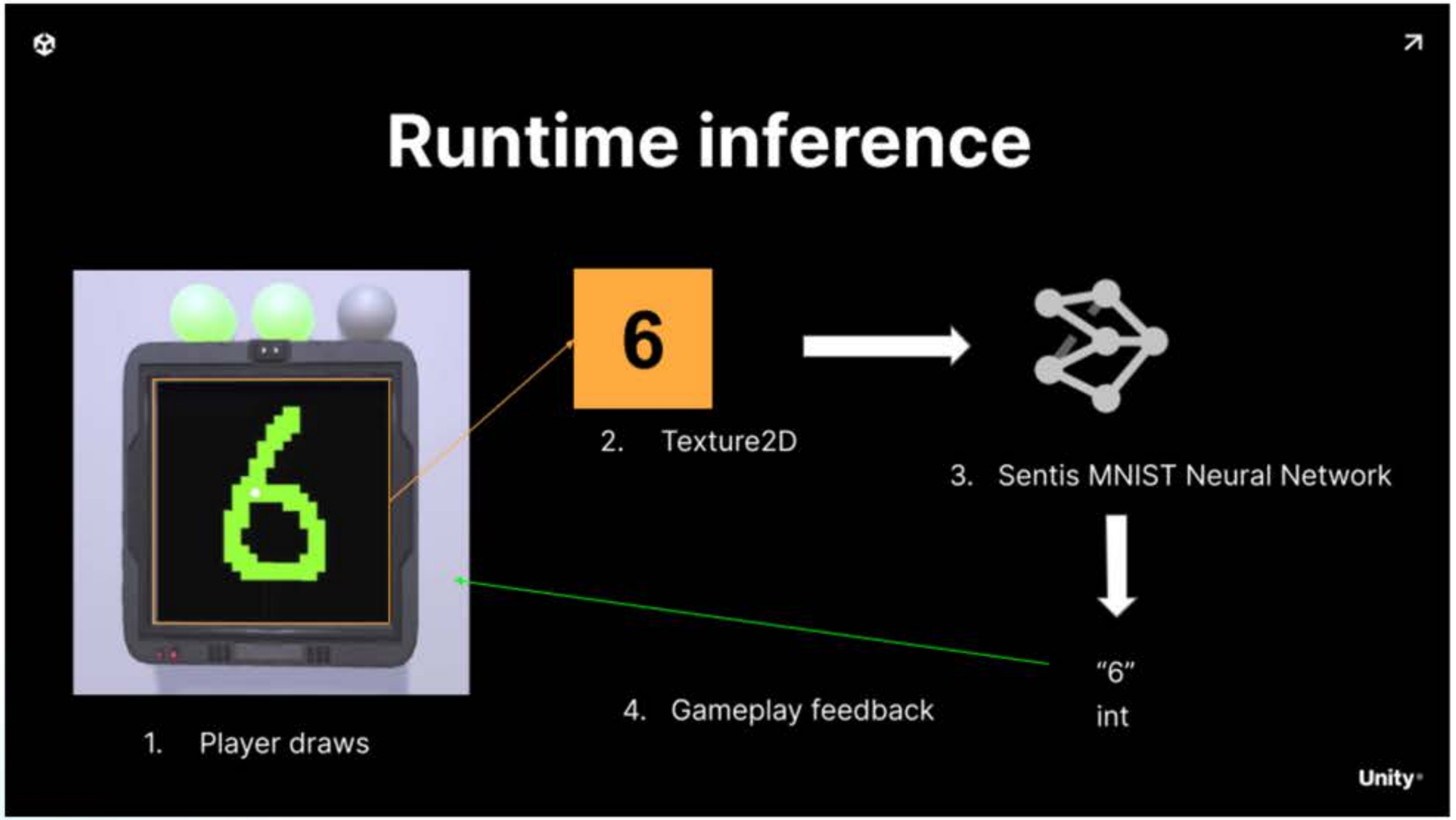
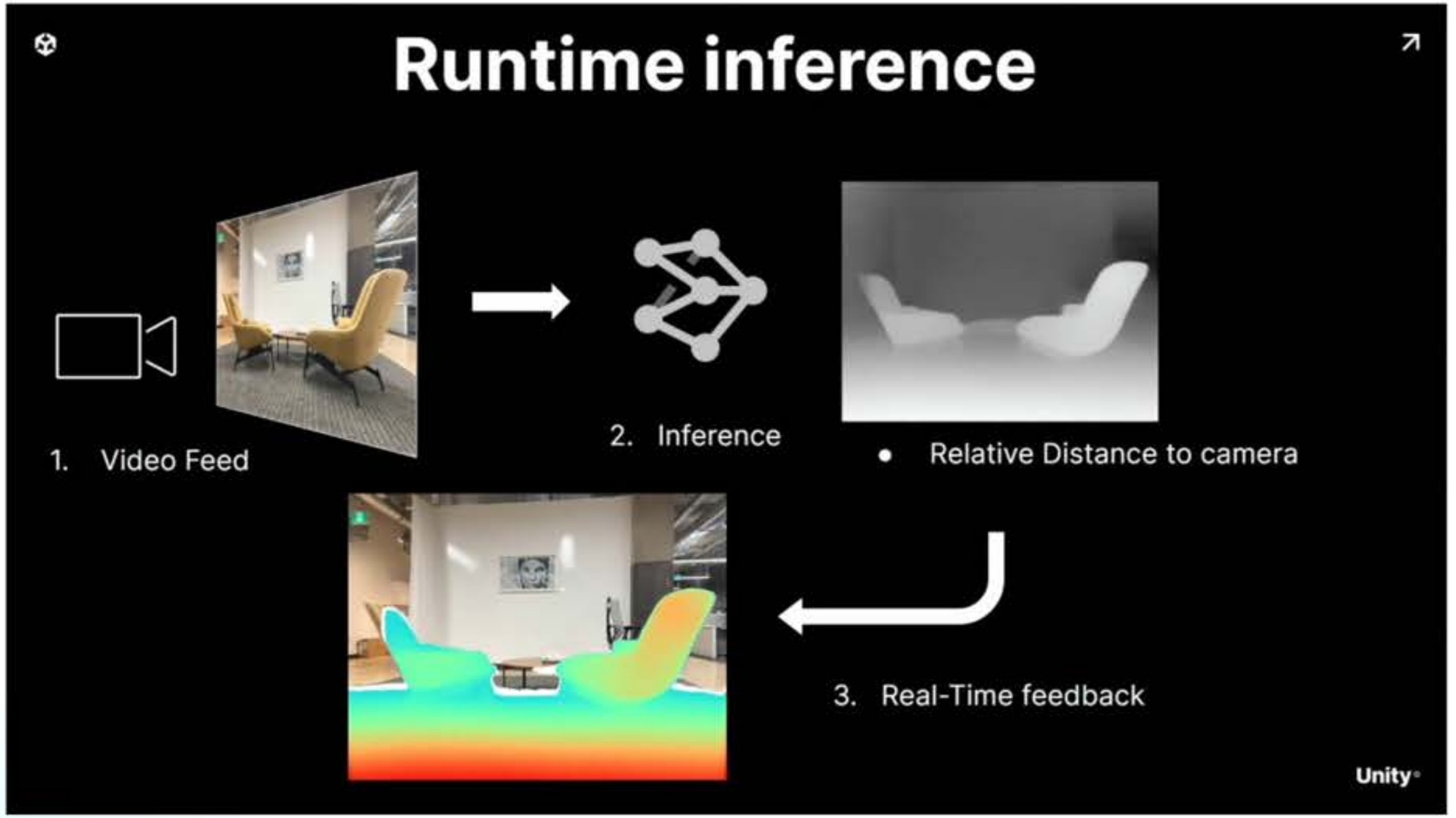


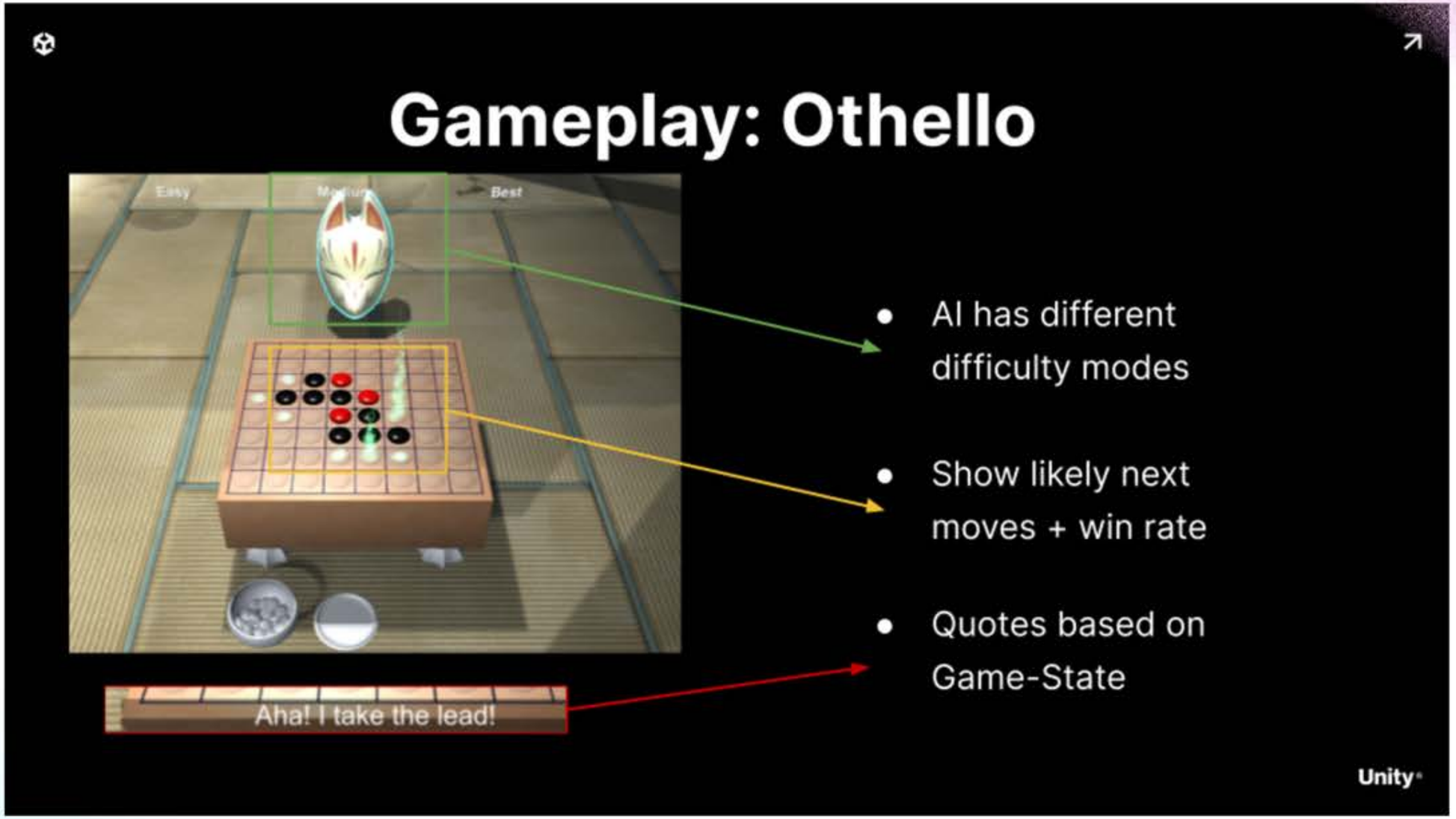
Image Classification



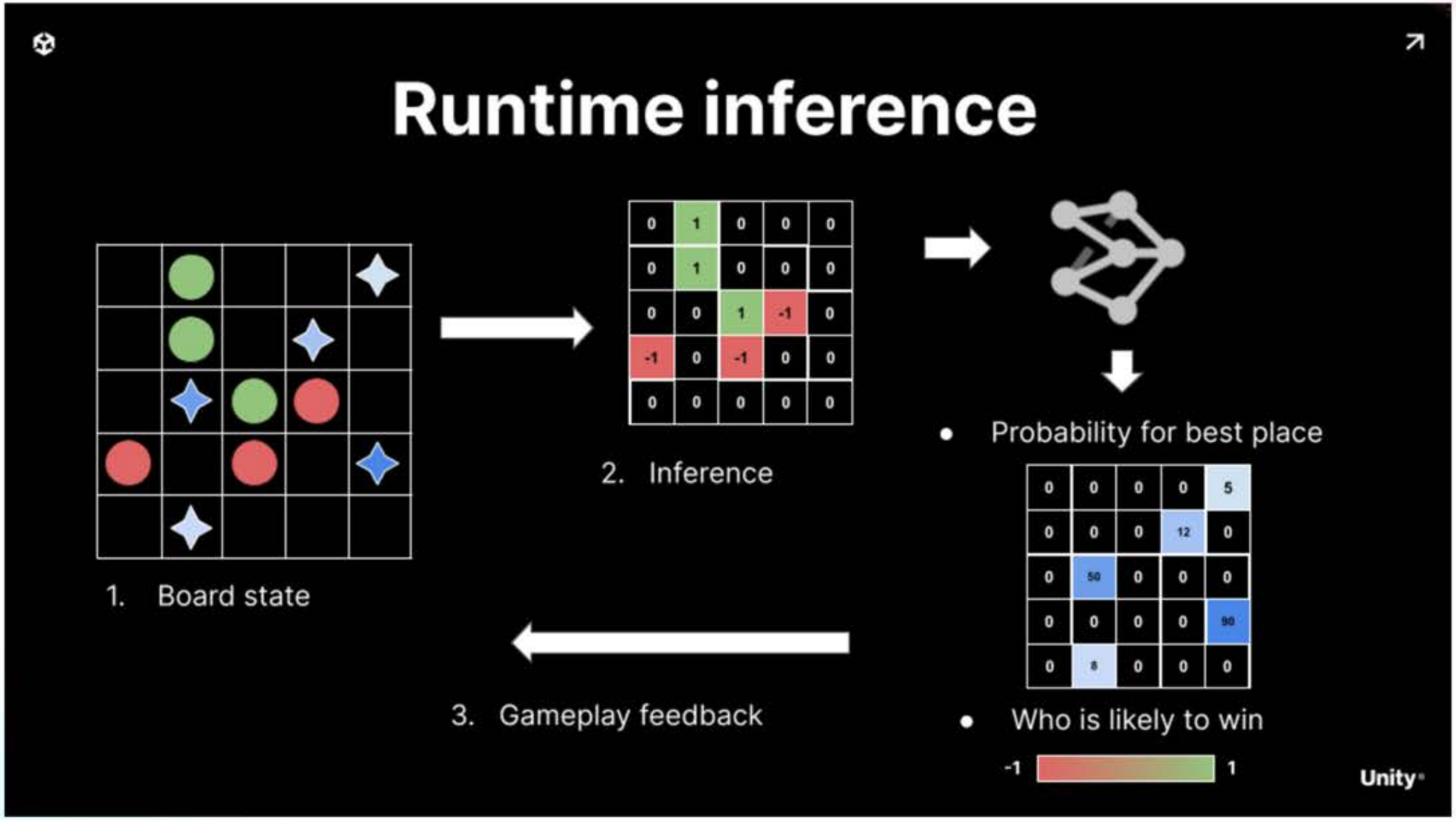
Depth Estimation



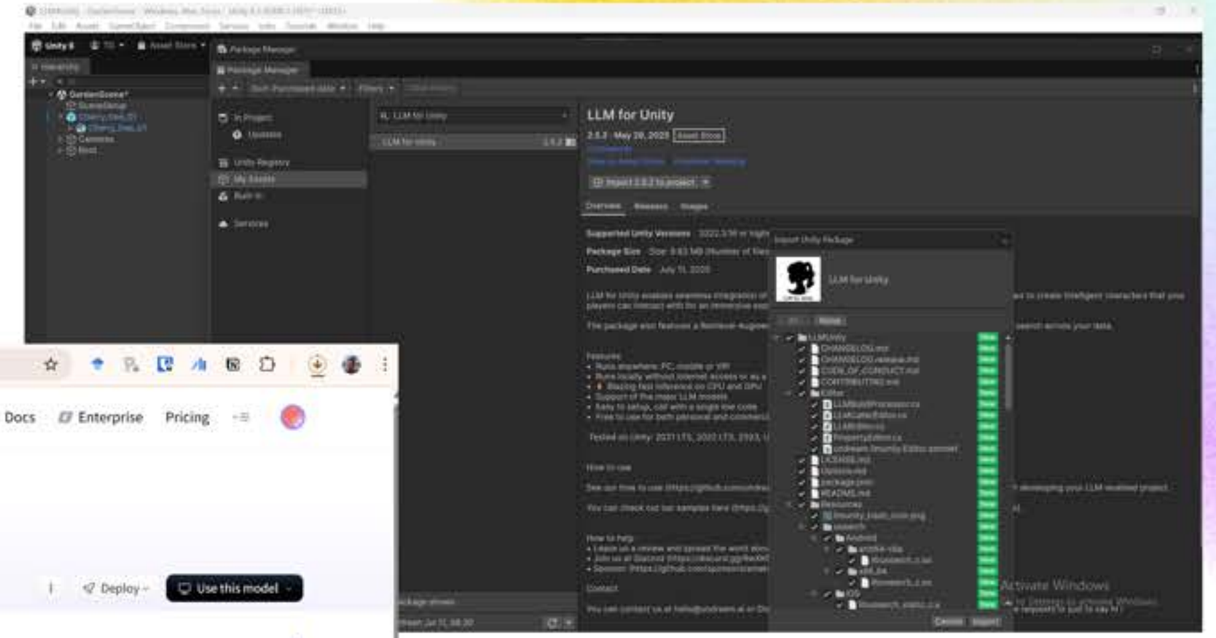
Board Game

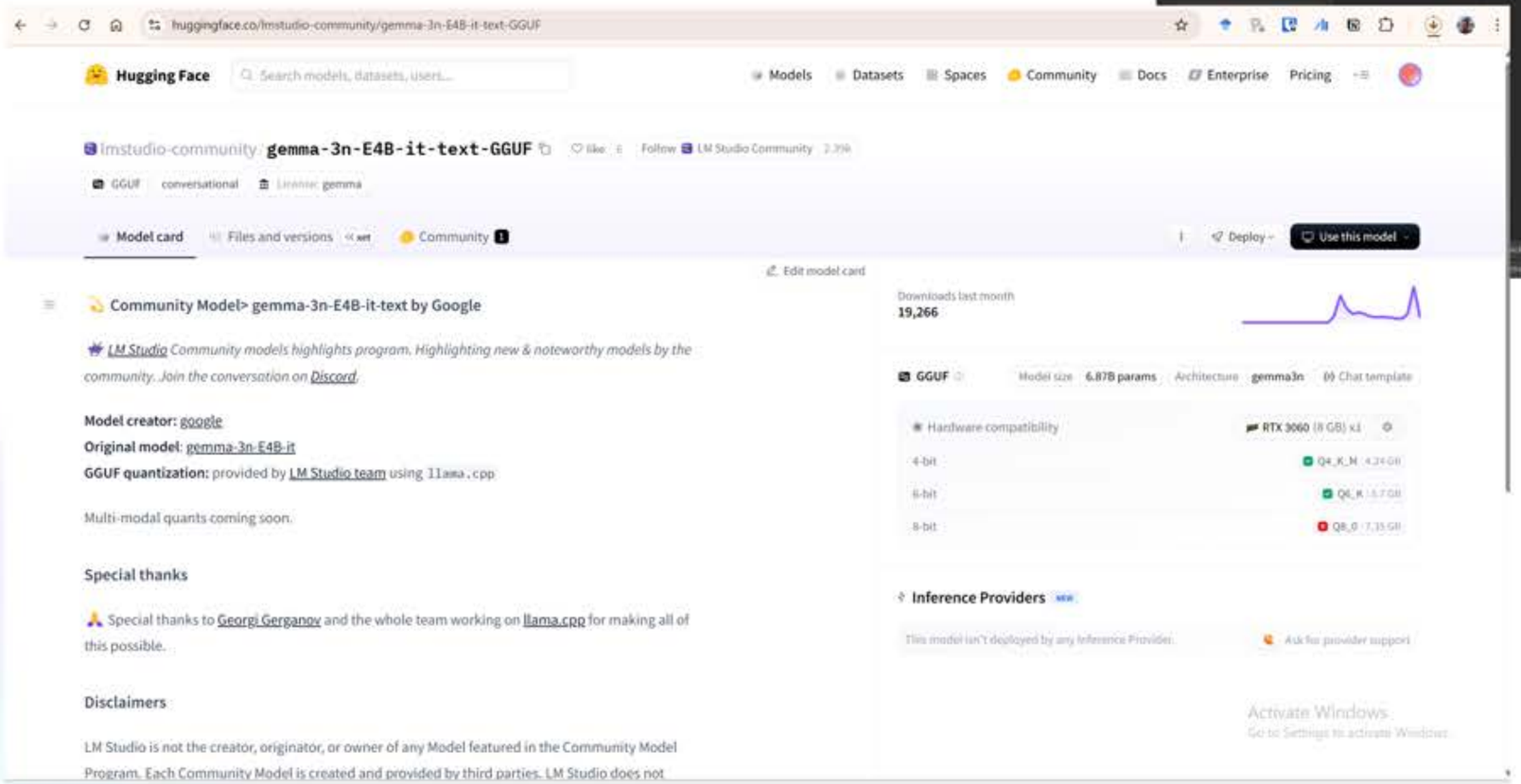

Gameplay: Othello
Easy
Best
Aha! I take the lead!
AI has different difficulty modes
Show likely next moves + win rate
Quotes based on Game-State
Unity

Board Game


Runtime inference
1. Board state
2. Inference
3. Gameplay feedback
• Probability for best place
• Who is likely to win
-1
1
Unity

LLM for Unity





🚀 Best Practices for AI in XR

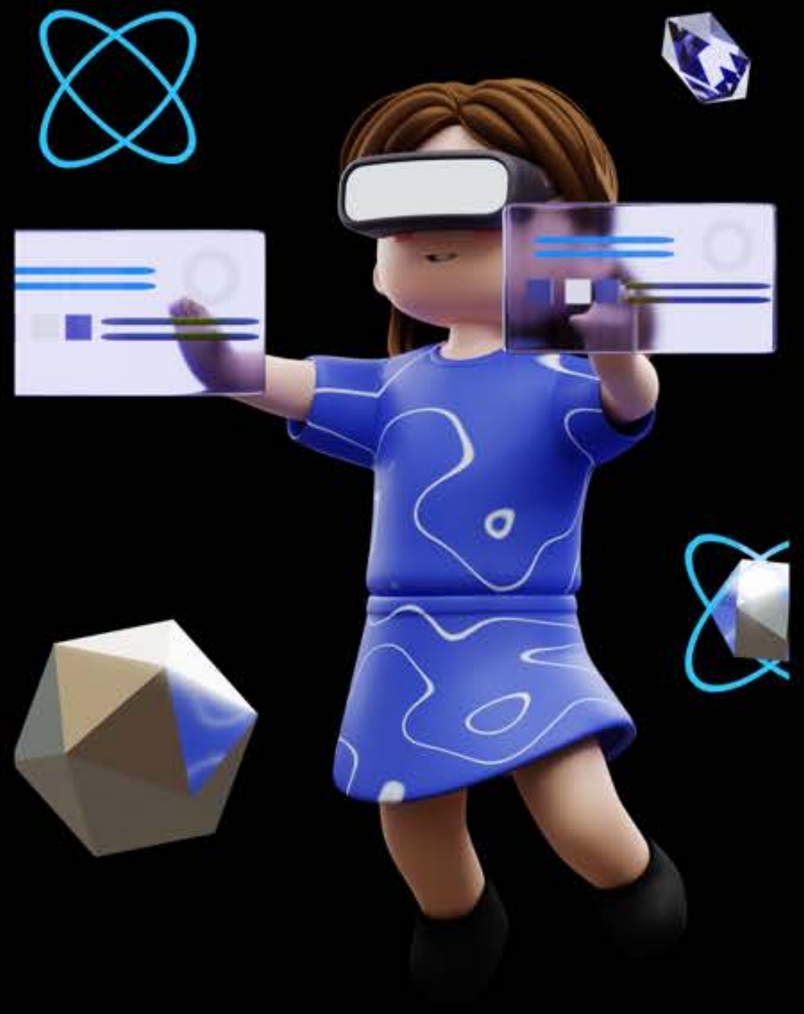
1. Optimize for Performance
    a. Use lightweight models (e.g., Tiny YOLO, MobileNet, Gemma3n) for AR/Mobile devices.
    b. Limit inference frequency (not every frame—use intervals).
2. Prioritize Latency & Responsiveness
    a. Always test response times for cloud APIs.
    b. Use loading indicators 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