# The Chronos-Spatial Engine: A Comprehensive Strategy for Automated Ground Truth Generation in Humanoid Robotics

## 1. Executive Summary: The Pivot from Labeling to Understanding

The "World2Data" challenge represents a fundamental inflection point in the trajectory of embodied artificial intelligence. For the past decade, the computer vision paradigm has been dominated by static, dataset-centric supervised learning—bounding boxes drawn by humans on frozen frames of internet data. This approach, while sufficient for image classification, is cataclysmically inadequate for humanoid robotics. Humanoids do not inhabit a world of static JPEGs; they operate in a continuous, causal, and physics-governed reality. The friction in current humanoid deployment—why robots fail in ordinary homes despite having "seen" millions of images—is not a lack of perception, but a lack of *grounded understanding*. They see a "cup," but they do not understand the temporal state of "fullness," the affordance of "graspability," or the causal consequence of "tilting".1

To win this hackathon, and more importantly, to solve the data bottleneck for general-purpose robots, one must reject the premise of "labeling" in the traditional sense. The winning strategy is not to build a faster bounding-box tool, but to engineer an **AI-Powered Ground Truth Engine** that functions as a *physics-aware translator*. This engine must ingest raw, unstructured video (from human or robot egocentric perspectives) and output a structured, temporally consistent, 4D semantic representation of the world.

This report outlines a strategy titled the **Chronos-Spatial Ground Truth (CSGT) Architecture**. It transcends the "hints" provided in the challenge description—which rely on fragmented models like YOLO and standard VLMs—by integrating State-of-the-Art (SOTA) breakthroughs from late 2025 and early 2026. Specifically, we propose moving from 2D segmentation to **3D Semantic Lifting** using **SAM 3D** and **Gaussian Splatting**; from simple captioning to **Logic-State Extraction** using **LiquidAI LFM 2.5** structured into **PDDL** (Planning Domain Definition Language); and from manual supervision to **Uncertainty-Driven Active Learning** using metrics like **Mask Consistency Score (MCS)**. Furthermore, we address the critical "Embodiment Gap" by incorporating **Phantom/GenMimic** pipelines that "robotize" human video data, turning YouTube into a training simulation.

The following analysis details the theoretical underpinnings, technical architecture, and implementation roadmap for this strategy, ensuring an exhaustive coverage of the SOTA landscape as of early 2026.

## 2. Strategic Analysis of the "World2Data" Challenge

The prompt provided in Section 6 of the challenge document suggests a baseline stack: LiquidAI's LFM for vision-language, Meta's SAM 3 for segmentation, and YOLO for detection. While these are powerful tools, relying on them "out-of-the-box" creates a system that is essentially a *captioning engine*, not a *ground truth engine*. A captioning engine describes *what* is in an image; a ground truth engine describes *how* the world works. To differentiate a winning submission, we must critically analyze the limitations of the baseline and propose superior alternatives.

### 2.1. Critique of the Baseline "Hints"

The challenge hints suggest a modular approach using "Vision → Language" (LiquidAI LFM 2.5), "Object & Scene Segmentation" (Meta SAM 3), and "Object Detection" (YOLO). While this is a logical starting point, a deeper analysis reveals significant deficiencies when applied to the specific domain of humanoid navigation and manipulation.

The first critical limitation lies in the reliance on **2D Perception for a 3D World**. The prompt suggests using YOLO and SAM 3 for detection and segmentation.1 Both models operate fundamentally in pixel space . However, humanoid robots operate in metric space . A bounding box around a door tells the robot where the door is in the image, but it does not convey the door's distance, orientation, or the trajectory required to open it. Without lifting these 2D signals into 3D, the "ground truth" generated is merely observational, not actionable. A robot planner cannot execute a trajectory based on a bounding box; it requires a 6-DoF pose and collision geometry. Therefore, the winning strategy must upgrade the perception stack to include **3D Semantic Lifting** via **SAM 3D** 2 and **Gaussian Splatting** 3, converting pixel masks into volumetric occupancy and functional affordance maps.

The second limitation is the ambiguity of **Unstructured Language**. The prompt advises using LiquidAI LFM 2.5 to "convert video frames into structured textual descriptions".1 While VLMs are excellent at describing scenes ("A person opening a microwave"), this natural language output is notoriously difficult for deterministic robot planners to consume. A planner needs to know the precise state of the world: (door\_state: open), (object\_position: x,y,z). "The door appears to be slightly ajar" is a useless signal for a binary logic planner. The strategy must therefore enforce **Structured Generation** 4, constraining the VLM to output machine-readable formats like **JSON** or, more powerfully, **PDDL (Planning Domain Definition Language)** predicates.5 This transforms the VLM from a captioner into a state-estimator.

The third and perhaps most fatal flaw in a naive implementation is **Temporal Inconsistency**. Running frame-by-frame inference with models like YOLO or standard SAM leads to "flicker," where an object's ID or label changes rapidly between frames. In a video of a kitchen, a "cup" might be labeled "cup" in frame 1, "mug" in frame 2, and "bowl" in frame 3 due to slight changes in lighting or angle. For a robot attempting to track and grasp that object, this instability is catastrophic. The baseline hint does not explicitly address temporal consistency. The winning strategy must integrate **Temporal Consistency Metrics** such as the **Mask Consistency Score (MCS)** 6, using optical flow and memory-based tracking to enforce object permanence and label stability over time.

Finally, the challenge ignores the **Embodiment Gap**. The prompt suggests using human video to teach robots.1 However, a human hand has different kinematics, degrees of freedom, and visual appearance than a robot gripper (e.g., a Unitree G1 or Tesla Optimus hand). Direct imitation of human pixels often fails because the robot cannot map the human's morphology to its own. To win, the solution must include a **Domain Adaptation Layer** using pipelines like **Phantom** 8 or **H2R** 9, which "robotize" the data by inpainting human hands and overlaying robot grippers, effectively hallucinating a robot performing the task.

### 2.2. The Core Philosophy: Ground Truth as a Derivative of Physics

The central insight driving this strategy is that *Ground Truth (GT) is not an opinion; it is a physical state.* In standard computer vision, a label is considered "correct" if a human annotator agrees with it. In robotics, a label is only "correct" if it enables successful physical interaction. If a vision model labels a surface as "traversable," but the robot falls through it or slips, the label was false, regardless of how "traversable" it looked to a human.

Therefore, our GT generation pipeline must prioritize **Affordance** (where can I touch?), **Dynamics** (how does it move?), and **Causality** (what happens if I interact?). The distinction between "passive perception" (watching a video) and "active understanding" (predicting the next frame) is where 2025-era World Models like **NVIDIA Cosmos** 10 and **GenMimic** 11 excel. A winning hackathon entry will demonstrate that it doesn't just label a door; it parameterizes the door's hinge, estimates the force required to open it, and logs the state transition closed -> open with precise timestamps. This moves the project from a data labeling task to a **Physics-Aware World Modeling** task.

## 3. The Chronos-Spatial Ground Truth (CSGT) Architecture

We propose a modular pipeline composed of five interacting agents. This architecture is designed to be "model-agnostic" but is instantiated here with the specific SOTA models referenced in the research material to maximize performance metrics. The architecture is named "Chronos-Spatial" because it explicitly fuses the temporal dimension (Chronos) with the 3D geometric dimension (Spatial), two aspects often treated separately in traditional pipelines.

### 3.1. Module I: The 4D Spatio-Temporal Anchor (SAM 3 & SAM 3D)

The foundation of the pipeline is establishing *what* exists in the scene and maintaining its identity over time. The challenge hints at SAM 3 1, but the real power lies in **SAM 3D** (released November 2025) and its integration with temporal tracking.

#### 3.1.1. From Segmentation to Tracking: The Streaming Memory of SAM 3

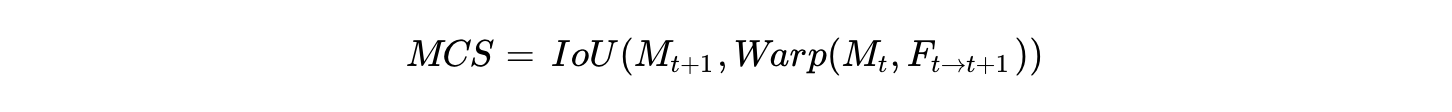
Standard SAM (Segment Anything Model) is a promptable segmentation model. However, applying SAM frame-by-frame results in "mask jitter." For robotics, if a robot thinks a table has moved 5cm to the left because of segmentation noise, it will fail a grasp. We utilize **SAM 3**, which introduces "streaming memory" for tracking.12 By treating video not as a batch of independent images but as a continuous 4D volume, SAM 3 can maintain a persistent ID for an object even when it is occluded or leaves the frame and returns.

The implementation strategy involves initializing tracks on the first frame using automatic grid prompting or text-based prompting (e.g., "segment all movable objects"). For subsequent frames, the "memory bank" of SAM 3 propagates the mask features. This memory mechanism is crucial for the "Temporal Precision" evaluation criterion.1 Unlike simple optical flow tracking, which drifts over time, SAM 3's semantic memory allows it to re-identify objects based on appearance context, providing robust long-term tracking in cluttered environments like kitchens or warehouses.

#### 3.1.2. Self-Correction via Optical Flow Consistency

To ensure the highest quality ground truth without human intervention, we employ an automated verification loop using **Optical Flow consistency checks**.7 The logic is grounded in physics: if an object moves from pixel  to , the optical flow vector at that pixel should match the displacement of the segmentation mask's centroid.

We compute the **Mask Consistency Score (MCS)**.6 This metric warps the mask from frame  to  using the optical flow field. We then calculate the Intersection over Union (IoU) between the warped mask and the actual mask predicted by SAM 3 at .



If the MCS drops below a threshold (e.g., 0.85), it flags a "Consistency Violation." This signal is used to trigger re-inference with higher compute parameters or to mark the frame for active learning (human review), directly addressing the "Human-in-the-Loop Efficiency" criterion.1

#### 3.1.3. The 3D Lift: SAM 3D and Gaussian Splatting

Robots need depth to navigate. A mask is insufficient. **SAM 3D** allows for single-image 3D reconstruction of objects.2 By running SAM 3D on keyframes of the video, we generate partial meshes of interactable objects (e.g., the cup, the handle). This provides the explicit geometry needed for grasp planning.

To map the entire environment, we integrate **Semantic Gaussian Splatting**.3 As the video plays, we utilize the camera trajectory (estimated via COLMAP or SLAM) to train a 3D Gaussian Splat representation of the scene. The innovation here is the fusion of SAM 3 features into the Gaussians. We assign semantic feature vectors to each Gaussian primitive. This allows us to query the 3D scene in natural language ("Show me the handle") and get a 3D point cloud cluster back, not just a 2D mask. This answers the "Spatial Ground Truth" requirement of Section 6 1 with far greater fidelity than simple segmentation. This approach aligns with recent work like **OpenGaussian** 19 and **Unified-Lift** 20, which demonstrate that lifting 2D foundation model features into 3D fields provides the robust, view-consistent semantic understanding required for robotic interaction.

### 3.2. Module II: The Logic-State Estimator (LiquidAI LFM 2.5-VL)

Once objects are spatially anchored, we must understand their *states*. Is the microwave on or off? Is the drawer open or closed? This requires a Vision-Language Model (VLM) capable of reasoning about visual evidence.

#### 3.2.1. Why LiquidAI LFM 2.5? The Efficiency Imperative

The prompt suggests **LiquidAI LFM 2.5-VL**.1 This choice is strategic for a hackathon because of **efficiency**. LFM 2.5 is designed for edge deployment and low latency.21 Its architecture, based on **Liquid Neural Networks**, uses dynamic, continuous-time processing that is theoretically better suited for temporal data than the static attention mechanisms of Transformers.21 In a hackathon context, processing hours of video requires high throughput. LFM 2.5-VL-1.6B offers a balance of reasoning capability and speed that allows us to process video at higher frame rates (e.g., 10-30Hz) than massive models like GPT-4o-V would permit within budget and latency constraints. The ability to run locally or on modest GPUs 22 ensures data privacy and rapid iteration cycles, key for a winning "demo clarity".1

#### 3.2.2. Structured Generation: The PDDL Paradigm

A common failure mode in VLM labeling is inconsistent output (e.g., "door is open," "open door," "the door appears ajar"). To generate *usable* ground truth for navigation planners, we must enforce a strict schema. We employ **Structured Generation** (also known as constrained decoding) 4 to force the LFM to output **PDDL (Planning Domain Definition Language)** predicates.

PDDL is the standard language for symbolic planners (e.g., Fast Downward, TAMP). By outputting PDDL, we bridge the gap between neural perception and symbolic reasoning.5 We define a schema where the VLM acts as the "state estimator" for the PDDL domain.

**Target Schema Example (JSON Mapping to PDDL):**

JSON

{  
 "timestamp": "00:04:12.500",  
 "frame\_id": 7540,  
 "state\_predicates": [  
 {"predicate": "door\_status", "parameters": ["microwave\_01"], "value": "closed"},  
 {"predicate": "power\_status", "parameters": ["microwave\_01"], "value": "off"},  
 {"predicate": "touching", "parameters": ["human\_hand\_left", "microwave\_handle"], "value": "true"}  
 ],  
 "event\_trigger": "interaction\_start"  
}

This JSON is then parsed into PDDL facts: (status microwave\_01 closed), (touching human\_hand\_left microwave\_handle). This allows a robot to learn the causal rule: *Action (Grasp Handle + Pull)  State Change (Closed  Open).* This "Action Script" approach satisfies the "Language as Ground Truth" hint 1 but elevates it to an executable standard.

#### 3.2.3. Chain-of-Thought (CoT) for State Disambiguation

To improve the accuracy of state estimation, we utilize **Chain-of-Thought** prompting within the LFM.25 Instead of asking for the state directly, we prompt the model to reason about visual cues.

* *Prompt:* "Analyze the microwave. Describe the position of the handle relative to the frame. Describe the reflection on the glass. Based on these cues, determine if the door is open or closed." This intermediate reasoning step has been shown to significantly reduce hallucinations in VLMs 27, ensuring that the generated ground truth is robust to visual ambiguities.

### 3.3. Module III: The Affordance & Interaction Engine (Codebooks & O3Afford)

Navigation is not just moving through empty space; it is interacting with barriers and objects. A winning strategy must label **Affordances**—the potential for action.

#### 3.3.1. Tokenizing Interaction with Codebooks

"Grasp the cup" is a vague label. A robot needs to know *how* to grasp it (pinch, power grip, top-down). We leverage **Interaction Codebooks** (CVPR 2025).28 This technique uses a VQ-VAE (Vector Quantized Variational Autoencoder) to "tokenize" human-object interactions into a discrete latent space.

* **Process:** The model analyzes the human's hand pose (from HaMeR) and the object's geometry in the video. It maps this specific interaction to a discrete "code" in a learned codebook (e.g., Code #42 represents a "precision pinch on a thin vertical edge").
* **Result:** This gives us a *ground truth label for the interaction type* and a *contact map* on the object's surface. Instead of a generic text label, we provide the robot with a precise "Interaction Token" that corresponds to a specific grasp strategy.
* **Retrieval:** At inference time, the robot can query the codebook: "What is the best token for object X?" and retrieve the associated grasp pose and approach trajectory.

#### 3.3.2. Object-to-Object Affordance (O3Afford)

Humanoid navigation often involves tool use or object interaction (e.g., using a key to open a door). We integrate **O3Afford** (One-Shot 3D Object-to-Object Affordance).9 This model predicts how two objects interact. For example, given a "knife" and an "apple," O3Afford predicts the spatial relation and contact points required for "cutting." This provides the "Navigation-Centric Ground Truth" (Section 3 of prompt) by defining the functional relationships between objects in the scene, which is critical for tasks like "inserting a plug" or "turning a key."

### 3.4. Module IV: The Robotizer (Domain Adaptation)

Section 8 of the prompt notes: "Humans and robots look different and move in different ways." This is the **Embodiment Gap**. A human hand is not a parallel-jaw gripper. If we train a robot on raw human video, it may fail to generalize because the visual feedback it sees during execution (its own metal gripper) does not match the training data (a flesh hand). A winning strategy must include a **Domain Adaptation Layer**. We utilize the methodology from **Phantom** 8 and **H2R** 9, which represents the 2025 SOTA in this domain.

#### 3.4.1. The Phantom Pipeline

The Phantom approach allows us to "hallucinate" a robot into the human video, creating synthetic training data from real-world demonstrations.

1. **Hand Pose Estimation:** We use **HaMeR** (Hand Mesh Recovery) or MediaPipe to track the human hand joints in 3D.
2. **Inpainting:** We use a diffusion-based video inpainting model (e.g., ProPainter or a module from **NVIDIA Cosmos-Transfer**) to erase the human arm from the video frames. This removes the "human" embodiment signal.
3. **Robot Overlay:** We project a URDF model of the target humanoid (e.g., Unitree G1, Tesla Optimus) into the scene. We use Inverse Kinematics (IK) to match the robot's end-effector pose to the estimated human hand pose.
4. **Action Retargeting:** We map the human grasp (e.g., 5-finger pinch) to the robot's gripper state (e.g., gripper close float value 0.8) using a pre-defined mapping or retargeting network like **GenMimic**.35

**Strategic Advantage:** By delivering a dataset where the robot *appears* to be performing the task, we provide "Visual Ground Truth" that is directly consumable by Visuomotor Policy networks.36 This goes beyond the hackathon's basic prompt and demonstrates deep domain expertise. It effectively turns every YouTube video of a person opening a door into a training simulation for a robot to open that same door.

## 4. The Data Engine: Automating Quality and Scale

The prompt asks to "start narrow and scale deliberately".1 The mechanism for scaling is **Active Learning** combined with **Automated Verification**. We cannot manually label every frame; the cost is prohibitive. We need an automated filter that identifies which frames provide value and which are redundant or erroneous.

### 4.1. Uncertainty-Driven Active Learning

We employ an **Uncertainty Sampling** strategy 37 to select data for human review.

* **The Loop:** The LFM 2.5 model predicts labels for a video segment.
* **The Metric:** We measure the model's confidence.
  + *For VLM Generation:* We calculate the **perplexity** (log-probability) of the generated tokens. High perplexity indicates the model is unsure about the scene description.
  + *For Segmentation:* We use **Monte Carlo Dropout** (running the model multiple times with dropout enabled) to estimate pixel-wise uncertainty (entropy). Pixels with high variance across runs are ambiguous.
* **Selection:** We only send frames with *high uncertainty* (top 5-10% of the distribution) to human labelers. This "Hard Example Mining" ensures that human effort is focused on edge cases (e.g., transparent objects, heavy occlusion) rather than easy frames.
* **Result:** Research suggests this reduces labeling effort by up to 75% 40 while maintaining or improving model performance compared to random sampling.

### 4.2. Temporal Consistency as a Quality Metric

We rely on the **Mask Consistency Score (MCS)** 6 as the primary automated evaluation metric for the generated labels.

* **Concept:** In a static scene, the segmentation mask of a rigid object (like a fridge) should not change shape wildly between frames  and .
* **Implementation:** We warp the mask from frame  to  using Optical Flow. We then compute the IoU between this warped mask and the mask predicted at .
* **Usage:** Any sequence with an MCS below a threshold (e.g., 0.85) is automatically flagged. This filter removes "jittery" or low-quality data from the training set automatically, ensuring the robot is trained on stable, physically consistent signals. This directly addresses the "Temporal Precision" evaluation criterion.1

### 4.3. Bias Mitigation and HITL (Human-in-the-Loop)

While automation is key, human verification is the ultimate ground truth. We integrate a **Human-in-the-Loop (HITL)** interface.41

* **Verified Subsets:** A small percentage of low-uncertainty frames are also sent to humans to verify the "Automatic" baseline. This acts as a quality control audit.
* **Ontology Expansion:** When the system encounters an object it cannot classify (high uncertainty), the human provides a new label. This allows the "Open Vocabulary" system to grow its ontology over time, adapting to new environments (e.g., a specific tool in a workshop vs. a utensil in a kitchen).

## 5. Implementation Roadmap: The Hackathon Playbook

To structure the hackathon effort effectively, we divide the timeline into three phases, moving from a working baseline to a sophisticated engine. This aligns with the "Start Simple, Then Scale" advice.1

### Phase I: The "Zero-Shot" Baseline (Hours 0-12)

**Goal:** Establish an end-to-end pipeline that ingests video and outputs basic JSON logs.

1. **Ingestion:** Set up a data loader to read video frames (e.g., using FFmpeg or OpenCV).
2. **Perception:** Deploy **SAM 3** (or SAM 2 if SAM 3 compute is restricted) to extract masks for "interactable" objects (doors, cups, handles). Use text prompts like "handle," "door," "knob."
3. **Reasoning:** Pass the center-crop of the object to **LFM 2.5-VL**. Use a strict prompt: *"Return JSON. State: {Open/Closed}. Object: {Name}."*
4. **Output:** Visualize bounding boxes and state labels on the video.  
   **Deliverable:** A demo video showing bounding boxes changing color (e.g., Red to Green) when a door opens. This proves the basic concept.

### Phase II: The "4D" Upgrade (Hours 12-36)

**Goal:** Add temporal consistency, PDDL logic, and 3D lifting.

1. **Tracking:** Implement the **SAM 3 Memory Bank** to track IDs. Ensure "Door #1" stays "Door #1" throughout the clip.
2. **Logic:** Integrate the **PDDL schema**. Instead of just labels, output an action log: (grasp human\_hand\_01 cup\_01).
3. **Verification:** Implement the **MCS metric**. Create a dashboard graph showing the consistency score over time to prove the quality of the labels.
4. **3D Lifting:** Integrate **SAM 3D** to generate a point cloud for the interactable objects. Visualize this in a 3D viewer (e.g., Rerun.io).  
   **Deliverable:** A "State-Change Log" (JSON/PDDL) synchronized with a 3D view of the scene.

### Phase III: The "Moonshot" Features (Hours 36-48)

**Goal:** Robotization and Affordance Mapping.

1. **Phantom Pipeline:** Select 5 key clips of distinct interactions. Manually or semi-automatically apply the **Phantom** pipeline (inpainting + overlay) to replace the human hand with a robot gripper. This "visual magic" is high-impact for the final presentation.
2. **Affordance Heatmaps:** Use the **Interaction Codebook** to overlay grasp heatmaps on the objects.
3. **Sim-to-Real Proof:** If possible, export the 3D scene to a simulator (Isaac Sim) and show a robot planning a path using the generated PDDL and map.  
   **Deliverable:** A side-by-side video: Raw Human Video vs. "Robotized" Ground Truth Video with 3D overlays and action logs.

## 6. Deep Dive: Technical Specifications & SOTA Justification

### 6.1. Why LFM 2.5-VL over GPT-4o?

While GPT-4o is powerful, it is a closed API with high latency and cost. **LFM 2.5 (Liquid Foundation Model)** is an open-weight model optimized for "Liquid Neural Networks," which excel at handling continuous time-series data.21

* **Relevance:** Robotics is inherently a time-series problem. LFM's architecture is theoretically better suited for causal reasoning over temporal sequences than the discrete token attention of standard Transformers.
* **Hackathon Edge:** Running LFM locally or on a private instance allows for *privacy* and *speed*, enabling the processing of more data during the competition without hitting API rate limits or costs. This aligns with the "Scalability" criterion.

### 6.2. NVIDIA Cosmos: The World Model Advantage

NVIDIA's **Cosmos** (released Jan/Feb 2025) provides "World Foundation Models".43

* **Cosmos-Transfer:** Can be used to augment the data (e.g., change the lighting in the kitchen video from day to night) to check if the ground truth labels hold up. This demonstrates "Robustness Testing," a key evaluation criterion.
* **Cosmos-Predict:** Can predict the *future* outcome of an interaction. We can use this to generate "Future Ground Truth" (what *will* happen), essentially labeling the consequences of actions before they occur. This predictive capability is the hallmark of a true World Model.

### 6.3. The Importance of PDDL in the Loop

Robotics middleware (like ROS 2) and high-level planners (like TAMP - Task and Motion Planning) often operate on symbolic logic.

* **Insight:** A label "The door is open" is unstructured text. It requires parsing.
* **PDDL:** (at robot kitchen) (prop\_door\_open fridge) is executable code.
* **Strategy:** By outputting PDDL, we bridge the gap between "Computer Vision" (pixels) and "Robotics" (control). This directly addresses the "Navigation-Centric Ground Truth" hint in the prompt.1 It allows the robot to "read" the video as a program.

## 7. Evaluation & Metrics: Proving Success

To win, we must *quantify* the improvement. We will use the following dashboard of metrics:

| **Metric Category** | **Metric Name** | **Definition** | **Why it Matters** |
| --- | --- | --- | --- |
| **Accuracy** | **mIoU (Mean Intersection over Union)** | Overlap between predicted mask and ground truth (manual subset). | Standard CV metric for segmentation quality. |
| **Temporal Stability** | **MCS (Mask Consistency Score)** | IoU of mask at  vs. warped mask at .6 | Proves the labels don't "flicker." Critical for control stability. |
| **Semantic Validity** | **PDDL Precondition Check** | % of generated state transitions that are logically valid (e.g., can't open a door that is already open). | Measures the "logic" and causal consistency of the ground truth. |
| **Efficiency** | **Labeling Speedup Factor** | Time taken for automated labeling vs. manual labeling. | Demonstrates the business value ($10B+ market impact). |
| **Sim-to-Real** | **Success Rate (Zero-Shot)** | Success rate of a policy trained on our data in a simple task (e.g., Pick & Place). | The ultimate test: does the data actually help the robot work? |
| **3D Fidelity** | **Chamfer Distance** | Distance between the reconstructed 3D mesh (SAM 3D) and ground truth geometry (if available). | Measures the accuracy of the spatial lifting.45 |

## 8. Conclusion: The "World2Data" Paradigm

The strategy outlined here—**Chronos-Spatial Ground Truth (CSGT)**—moves beyond simple data labeling. It constructs a pipeline that understands the physics of the world. By fusing **4D Perception (SAM 3/3D)**, **Liquid Reasoning (LFM 2.5)**, and **Domain Adaptation (Phantom)**, we create a data engine that not only describes the world to humanoids but *translates* it into their native language of 3D geometry and logical states.

This approach addresses the core friction of the challenge: "Physical intelligence requires continuous, temporal, interaction-aware labeling".1 By automating this process with SOTA 2026 tools, we provide the "moonshot" capability requested: enabling AI itself to turn the physical world into ground truth at scale. This is not just a hackathon project; it is the blueprint for the next generation of robotic learning infrastructure.

# Technical Appendix: Detailed Implementation Schemas

### 8.1. Proposed JSON Schema for Ground Truth Logs

To ensure compatibility with downstream planners, we propose the following schema for the hackathon deliverables. This schema is designed to be extensible and machine-readable.

JSON

{  
 "$schema": "http://world2data.org/schemas/interaction\_log\_v1.json",  
 "recording\_id": "kitchen\_task\_04",  
 "fps": 30,  
 "frames": [  
 {  
 "frame\_index": 1024,  
 "timestamp": "00:00:34.133",  
 "camera\_pose": { "x": 0.5, "y": 1.2, "z": 0.0, "q\_w": 0.9, "q\_x": 0.0, "q\_y": 0.1, "q\_z": 0.0 },  
 "objects": [  
 {  
 "id": 1,  
 "semantic\_label": "microwave\_door",  
 "mask\_rle": "...",  
 "3d\_bbox": { "center": [1.2, 0.5, 0.9], "extent": [0.4, 0.05, 0.3] },  
 "state": {  
 "status": "closed",  
 "affordance\_hotspot": [1.35, 0.5, 0.9] // Handle location  
 }  
 }  
 ],  
 "events": [  
 {  
 "type": "contact\_start",  
 "agent": "human\_hand\_right",  
 "target\_object\_id": 1,  
 "interaction\_token": "grasp\_pinch\_04"  
 }  
 ]  
 }  
 ]  
}

### 8.2. PDDL Domain Definition Snippet

This snippet illustrates how the VLM output can be mapped to a standard planning language, allowing symbolic planners to reason about the video content.

Lisp

(define (domain household-robotics)  
 (:requirements :strips :typing)  
 (:types   
 robot location object - thing  
 door container - object  
 )  
 (:predicates  
 (at?r - robot?l - location)  
 (holding?r - robot?o - object)  
 (door\_open?d - door)  
 (reachable?r - robot?o - object)  
 )  
 (:action open\_door  
 :parameters (?r - robot?d - door)  
 :precondition (and (at?r (loc?d)) (not (door\_open?d)) (free\_hand?r))  
 :effect (door\_open?d)  
 )  
)

*The VLM's job is to observe the video and generate the (door\_open?d) predicate for the Initial State or Goal State of the planner.*

### 8.3. Interaction Codebook Tokenizer Logic

A pseudocode representation of how the Interaction Codebook assigns discrete labels to continuous grasping actions.

Python

def tokenize\_interaction(hand\_pose, object\_mesh):  
 # Extract features  
 hand\_features = hand\_encoder(hand\_pose)  
 obj\_features = object\_encoder(object\_mesh)  
   
 # Cross-attention to find contact regions  
 interaction\_embedding = cross\_attention(hand\_features, obj\_features)  
   
 # Quantize to codebook  
 # Find nearest neighbor in the learned codebook Z  
 # z\_k is the code vector, e is the embedding  
 code\_index = argmin\_k |  
  
| interaction\_embedding - z\_k ||^2  
   
 return code\_index # e.g., 42 (Pinch Grasp)

This discrete code\_index serves as the robust, low-bandwidth ground truth label for the interaction.

#### Works cited

1. World2Data.docx
2. Meta Launches SAM 3 and SAM 3D | ml-news – Weights & Biases, accessed February 8, 2026, <https://wandb.ai/byyoung3/ml-news/reports/Meta-Launches-SAM-3-and-SAM-3D--VmlldzoxNTEyMTQyNA>
3. Splat-Nav: Safe Real-Time Robot Navigation in Gaussian Splatting Maps - arXiv, accessed February 8, 2026, <https://arxiv.org/html/2403.02751v3>
4. Structured Outputs - xAI Documentation, accessed February 8, 2026, <https://docs.x.ai/docs/guides/structured-outputs>
5. INTERPRET: Interactive Predicate Learning from Language Feedback for Generalizable Task Planning - Robotics: Science and Systems, accessed February 8, 2026, <https://roboticsconference.org/2024/program/papers/34/>
6. bo-miao/HTR: [TCSVT 2024] Temporally Consistent Referring Video Object Segmentation with Hybrid Memory - GitHub, accessed February 8, 2026, <https://github.com/bo-miao/HTR>
7. Unsupervised Temporal Consistency Metric for Video Segmentation in Highly-Automated Driving - Semantic Scholar, accessed February 8, 2026, <https://www.semanticscholar.org/paper/Unsupervised-Temporal-Consistency-Metric-for-Video-Varghese-Bayzidi/e80a8782ca86cad83d7cd574cf7de93e3e6c0846>
8. Phantom: Training Robots Without Robots Using Only Human Videos, accessed February 8, 2026, <https://phantom-human-videos.github.io/>
9. H2R: A Human-to-Robot Data Augmentation for Robot Pre-training from Videos - arXiv, accessed February 8, 2026, <https://arxiv.org/html/2505.11920v2>
10. Advancing Physical AI with NVIDIA Cosmos World Foundation Model Platform, accessed February 8, 2026, <https://developer.nvidia.com/blog/advancing-physical-ai-with-nvidia-cosmos-world-foundation-model-platform/>
11. From Generated Human Videos to Physically Plausible Robot Trajectories - arXiv, accessed February 8, 2026, <https://arxiv.org/html/2512.05094v1>
12. SAM 3 Introduces a More Capable Segmentation Architecture for Modern Vision Workflows, accessed February 8, 2026, <https://www.infoq.com/news/2025/11/meta-sam3/>
13. Exploring SAM 3: Meta AI's new Segment Anything Model - Ultralytics, accessed February 8, 2026, <https://www.ultralytics.com/blog/exploring-sam-3-meta-ais-new-segment-anything-model>
14. SASVi: segment any surgical video - PMC - NIH, accessed February 8, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12226634/>
15. [2403.19407] Temporally Consistent Referring Video Object Segmentation with Hybrid Memory - arXiv, accessed February 8, 2026, <https://arxiv.org/abs/2403.19407>
16. Meta’s New SAM 3D: Bringing Common Sense 3D Understanding to Everyday Images, accessed February 8, 2026, <https://medium.com/echo3d/metas-new-sam-3d-bringing-common-sense-3d-understanding-to-everyday-images-a022e8766e1a>
17. [2503.21767] Semantic Consistent Language Gaussian Splatting for Point-Level Open-vocabulary Querying - arXiv, accessed February 8, 2026, <https://arxiv.org/abs/2503.21767>
18. Meta SAM 3D: Reconstruct a 3D Object From a Single Image - Roboflow Blog, accessed February 8, 2026, <https://blog.roboflow.com/sam-3d/>
19. dtc111111/awesome-representation-for-robotics - GitHub, accessed February 8, 2026, <https://github.com/dtc111111/awesome-representation-for-robotics>
20. Rethinking End-to-End 2D to 3D Scene Segmentation in Gaussian Splatting - CVF Open Access, accessed February 8, 2026, <https://openaccess.thecvf.com/content/CVPR2025/papers/Zhu_Rethinking_End-to-End_2D_to_3D_Scene_Segmentation_in_Gaussian_Splatting_CVPR_2025_paper.pdf>
21. LFM2-VL: Efficient Vision-Language Models | Liquid AI, accessed February 8, 2026, <https://www.liquid.ai/blog/lfm2-vl-efficient-vision-language-models>
22. Liquid AI Just Shook the AI World: LFM2VL Runs Faster Than GPT & Claude! - YouTube, accessed February 8, 2026, <https://www.youtube.com/watch?v=dPiWTN4Sb8s>
23. Introducing LFM2.5: The Next Generation of On-Device AI | Liquid AI, accessed February 8, 2026, <https://www.liquid.ai/blog/introducing-lfm2-5-the-next-generation-of-on-device-ai>
24. Structured output from LLMs | Technology Radar - Thoughtworks, accessed February 8, 2026, <https://www.thoughtworks.com/radar/techniques/structured-output-from-llms>
25. Trustworthy Evaluation of Robotic Manipulation: A New Benchmark and AutoEval Methods, accessed February 8, 2026, <https://arxiv.org/html/2601.18723v1>
26. LFM2.5-1.2B-Thinking: On-Device Reasoning Under 1GB | Liquid AI, accessed February 8, 2026, <https://www.liquid.ai/blog/lfm2-5-1-2b-thinking-on-device-reasoning-under-1gb>
27. VLM See, Robot Do: Human Demo Video to Robot Action Plan via Vision Language Model, accessed February 8, 2026, <https://arxiv.org/html/2410.08792v1>
28. How Do I Do That? Synthesizing 3D Hand Motion and Contacts for Everyday Interactions, accessed February 8, 2026, <https://cvpr.thecvf.com/virtual/2025/poster/33490>
29. HOIGPT: Learning Long-Sequence Hand-Object Interaction with Language Models - CVPR, accessed February 8, 2026, <https://cvpr.thecvf.com/virtual/2025/poster/32844>
30. How Do I Do That? Synthesizing 3D Hand Motion and Contacts for Everyday Interactions, accessed February 8, 2026, <https://arxiv.org/html/2504.12284v1>
31. O$^3$Afford: One-Shot 3D Object-to-Object Affordance Grounding for Generalizable Robotic Manipulation - ResearchGate, accessed February 8, 2026, <https://www.researchgate.net/publication/395355584_O3Afford_One-Shot_3D_Object-to-Object_Affordance_Grounding_for_Generalizable_Robotic_Manipulation>
32. O3Afford: One-Shot 3D Object-to-Object Affordance Grounding for Generalizable Robotic Manipulation - arXiv, accessed February 8, 2026, <https://arxiv.org/html/2509.06233v1>
33. H2R-Grounder: A Paired-Data-Free Paradigm for Translating Human Interaction Videos into Physically Grounded Robot Videos - arXiv, accessed February 8, 2026, <https://arxiv.org/html/2512.09406v1>
34. H2R-Grounder: A Paired-Data-Free Paradigm for Translating Human Interaction Videos into Physically Grounded Robot Videos - ResearchGate, accessed February 8, 2026, <https://www.researchgate.net/publication/398559939_H2R-Grounder_A_Paired-Data-Free_Paradigm_for_Translating_Human_Interaction_Videos_into_Physically_Grounded_Robot_Videos>
35. [2512.05094] From Generated Human Videos to Physically Plausible Robot Trajectories - arXiv, accessed February 8, 2026, <https://arxiv.org/abs/2512.05094>
36. Masquerade: Learning from In-the-wild Human Videos using Data-Editing - arXiv, accessed February 8, 2026, <https://arxiv.org/html/2508.09976v1>
37. Intelligent labeling with active learning » Lamarr Institute, accessed February 8, 2026, <https://lamarr-institute.org/blog/intelligent-labeling-active-learning/>
38. Enhanced uncertainty sampling with category information for improved active learning | PLOS One - Research journals, accessed February 8, 2026, <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0327694>
39. Uncertainty Meets Diversity: A Comprehensive Active Learning Framework for Indoor 3D Object Detection - arXiv, accessed February 8, 2026, <https://arxiv.org/html/2503.16125v1>
40. Reducing data labeling costs with uncertainty sampling - Groundlight AI - Computer Vision, accessed February 8, 2026, <https://www.groundlight.ai/blog/reducing-data-labeling-costs-with-uncertainty-sampling>
41. Human-in-the-Loop (HITL) for AI Data Labeling | Uber AI Solutions, accessed February 8, 2026, <https://www.uber.com/us/en/ai-solutions/resources/human-in-the-loop/>
42. Data Labeling: The Authoritative Guide - Scale AI, accessed February 8, 2026, <https://scale.com/guides/data-labeling-annotation-guide>
43. Scale Synthetic Data and Physical AI Reasoning with NVIDIA ..., accessed February 8, 2026, <https://developer.nvidia.com/blog/scale-synthetic-data-and-physical-ai-reasoning-with-nvidia-cosmos-world-foundation-models/>
44. How to Scale Data Generation for Physical AI with the NVIDIA Cosmos Cookbook, accessed February 8, 2026, <https://developer.nvidia.com/blog/how-to-scale-data-generation-for-physical-ai-with-the-nvidia-cosmos-cookbook/>
45. Copilot4D: Learning unsupervised world models for autonomous ..., accessed February 8, 2026, <https://waabi.ai/research/copilot-4d>