World Modeling with Probabilistic Structure Integration

Stanford NeuroAI Lab*

September 15, 2025

We present Probabilistic Structure Integration (**PSI**), a system for learning richly controllable and flexibly promptable world models from data. PSI consists of a three-step cycle. The first step, **P**robabilistic prediction, involves building a probabilistic graphical model Ψ of the data, in the form of a random-access autoregressive sequence model. Ψ supports a complete set of learned conditional distributions describing the dependence of any variables in the data on any others set of variables. In step 2, **S**tructure extraction, we show how to extract underlying low-dimensional properties in the data, corresponding to a diverse set of meaningful "intermediate structures", in a zero-shot fashion via causal inference on Ψ. Step 3, **I**ntegration, completes the cycle by converting these structures into new token types that are then continually mixed back into the training diet as conditioning signals and prediction targets. Each such cycle augments the capabilities of Ψ, both allowing it to model the underlying data better, and creating new control handles – akin to an LLM-like universal prompting language. We train an instance of Ψ on 1.4 trillion tokens of internet video data; we use it to perform a variety of useful video prediction and understanding inferences; we extract state-of-the-art optical flow, self-supervised depth and object segmentation; and we use these structures to support a full cycle of predictive improvements.
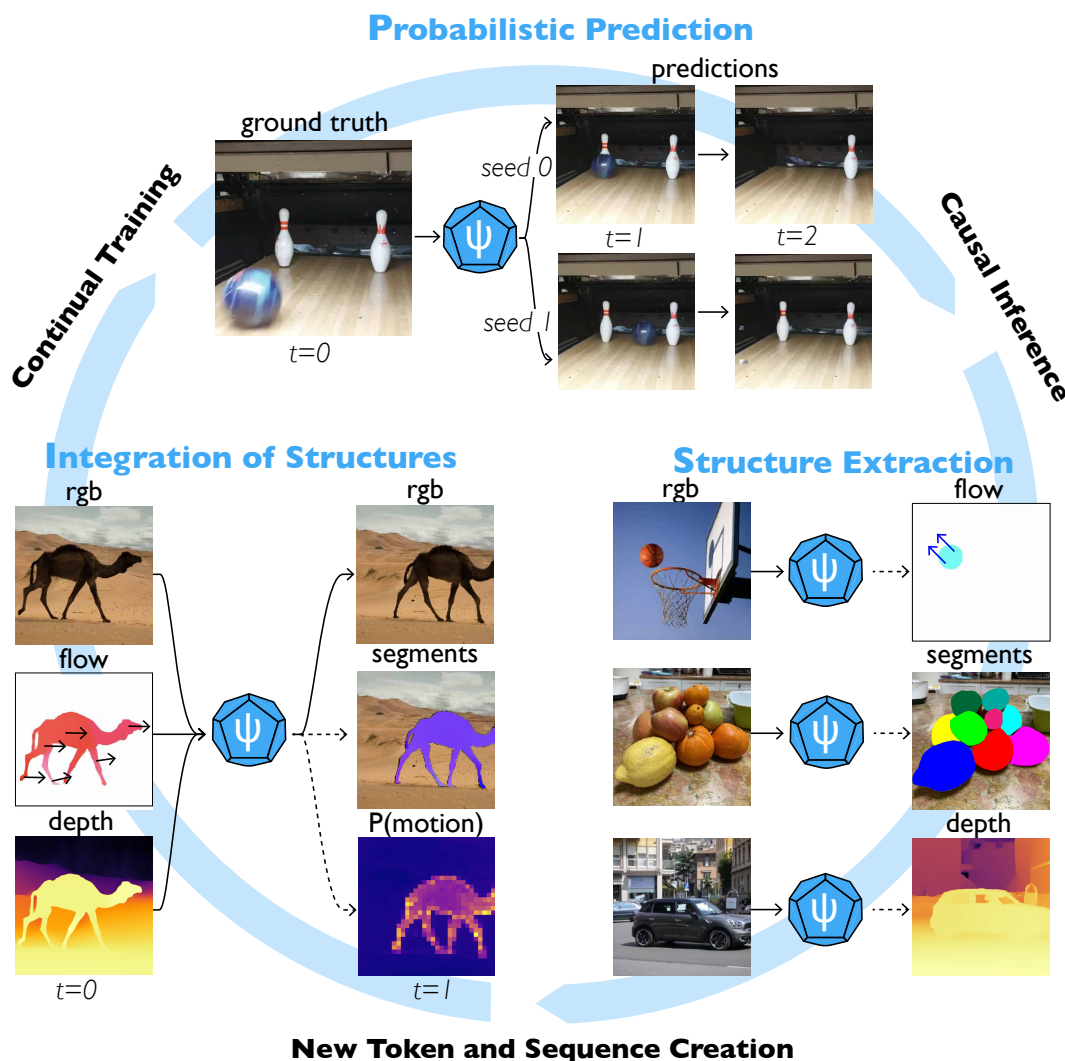
Figure 1: Probabilistic Structure Integration. We propose a new system for constructing self-improving world models consisting of a three-step cycle. First, a probabilistic predictor is trained on raw input data (e.g., RGB pixels). Second, the model is prompted to produce counterfactual generations, which isolate underlying structures in the data (e.g., optical flow, depth, segments). Third, those structures are integrated back into the predictor as both conditioning and prediction targets through continual training to improve its predictions and expand its capabilities.

Contents

Contributors to this work include: Klemen Kotar, Wanhee Lee*, Rahul Venkatesh*, Honglin Chen*,†, Daniel Bear†, Jared Watrous, Simon Kim, Khai Loong Aw, Lilian Naing Chen, Stefan Stojanov†, Kevin Feigelis†, Imran Thobani, Alex Durango, Khaled Jedoui, Atlas Kazemian, and Dan Yamins*. (*=core contributor, †=alum)

1   Introduction

The natural world is a complex system governed by a set of physical laws [1]. Intelligent agents can exploit this property by forming models of the world to help reason about causal relationships and plan actions that achieve complex goals [2, 3, 4]. However, despite significant progress in learned simulators [5, 6] and generative AI [7, 8], today's world models have some key limitations, including a lack of *rich controllability* and *flexible query-ability*. Unlike pure language models, which can in their own domain be prompted in almost arbitrarily flexible (and enormously fun) ways, the "API" for interacting with the non-linguistic portions of world models — e.g., the visual, motor, or memory components – is often coarse and awkward, relying on discrete "game-like" action sets [5] or short text prompts [9, 8] that very imperfectly capture the physical needs of real-world planning and control. In these settings, one cannot readily nudge an articulated limb, peel back an occluder, adjust the viewpoint, or manipulate specific objects – precluding the richness of interaction that enables complex behavior. Foundation models in non-language domains are also awkward to query to expose their underlying world understanding, relying on representational learning with supervised probes to extract any given feature of interest. In language, a single model with a unified prompting interface excels at both controllability and query-ability; in vision, the absence of such an interface forces us to build separate models for generative and discriminative tasks.

In this work, we propose **Probabilistic Structure Integration (PSI)**, a generic process for building richly controllable world models with the right interfaces for true interactivity – even when starting from low-level nonlinguistic raw input data. PSI consists of three steps that form a virtuous cycle. Step 1 involves the training of a **P**robabilistic predictor Ψ that answers flexible visual queries at a local, low level of detail–close to raw data. Step 2 involves extracting **S**tructures from the predictor Ψ, corresponding to natural intermediate features for understanding the world – effectively, new token types that transcend the low-level raw inputs. Step 3 involves **I**ntegrating those new token types back into the predictor Ψ, using them as both new conditioning sources and prediction targets. The PSI process can be iterated, with each cycle enlarging the model's control surfaces, enabling the construction of further new structures, and steadily increasing the predictive fidelity of the model itself. A key feature of PSI is that the model interface remains the same throughout the process, with no need for expanded architectures of complex new types as the model self-enriches.

The core mechanism of Step 1 is to have Ψ predict a complete set of conditional distributions of local variables in the world, conditioned on arbitrary subsets of other such local variables. For example, in the case of natural video — the main case we explore empirically in this work — local variables correspond to (properly tokenized) spatiotemporal patches of RGB values. Here, Ψ learns to predict the conditional distribution of any patch in a visual clip, conditioned on an arbitrary subset of other such patches. Because the world branches into multiple valid futures from any given observation, Ψ represents the full conditional distribution in each prediction rather than a single best guess, so it can both sample plausible outcomes and compute high-level scene statistics, such as conditional entropy. This interface supports a wide range of inference pathways about the scene: nudge a small area and see what else might change; reveal a few local observations and have the model synthesize coherent completions of the rest of the scene; or just perform unconditional predictions about the distributions over possible future states. In other words, the Ψ model is like a <u>learned probabilistic graphical model</u> over patch variables, in which any observation can condition any prediction. Learning a full probabilistic model of this kind may sound like a challenging problem: indeed, the key technical advance in Step 1 of PSI (see §2) is formulating the Ψ model so that it can be scalably learned from raw video data. Specifically, we show how to construct Ψ as a "random-access" autoregressive transformer that learns stepwise conditional distributions on many random orderings of the same sparse visual data. This formulation has the twin advantages: being extremely data efficient and as enabling us to repurpose the highly scalable modern LLM code infrastructure while avoiding task-specific heads or architectural forks.

Low-level variables can only go so far, however, in supporting complex world understanding and control goals. This is where the **S**tructure-extraction aspect of PSI comes in. Physically, structure extraction is based on the observation that interacting with the RGB-based Ψ reveals natural groupings that exist in a scene (e.g., pixels that move together) and thereby exposes lower-dimensional regularities in the underlying structure of

reality. From the probabilistic graphical model view, we can intervene on any subset of local variables and measure induced changes elsewhere. By designing manipulations that elicit correlated responses across nodes, we expose axes of covariance and thereby uncover the latent structures that cause those correlations. The core technical insight (see §3) is that such structures — sometimes called *intermediate* representations or *properties* — arise in a zero-shot fashion as causal inferences in Ψ, which in turn are comparisons between Ψ-predictions under factual and hypothetical interventions. Classical computer-vision quantities are concrete instances of these low-dimensional factors: *optical flow* is the causal effect of appearance on future position [10], *depth* arises from the causal relationship between viewpoint change and parallax [11], and *segments* identify elements whose pixels share common fate under motion [12]. With a sufficiently controllable predictor, these properties can be extracted zero-shot via counterfactual prompts. The idea of visual prompting was pioneered by Counterfactual World Modeling (CWM) [13] with regression predictors; in PSI's *distributional* setting, Ψ produces sharp, coherent, multimodal outcomes that cleanly isolate causal effects where regression collapses them into blurry averages.

Intermediate structures can double as concise interfaces for describing and analyzing scenes, reducing complexity while aligning with physical organization. This observation motivates the third PSI step, where we close the loop by **I**ntegrating the extracted structures back into the predictor Ψ. The core technical insight of this step is a very simple generic mechanism for new token-type integration. Essentially, this mechanism has three parts: (1) create the new tokens from outputs of step 2, using the same kind of tokenizer (and the same vocabulary) that works for RGB; (2) mix the new tokens into the middle of RGB token sequences, creating new mixed-type sequences for training to complement previous pure-type sequences; and (3) continue to train the same Ψ model, on a mixture of the old and new sequences, from wherever it was previously left off training as if nothing had changed in the training diet. Implementing this radically simple approach requires only a few key technical choices during implementation (as detailed in §4) — the main empirical fact is that it works well, akin to mixing new synthetic data into an LLM's diet mid-stream [14]. The same unified model can then, for example, predict flow from RGB, generate RGB conditioned on flow, or compose these capabilities with depth and segments in arbitrary combinations. Once integrated, each structure serves a dual role—as a conditioning signal and a prediction target—and, crucially, becomes a first-class *control surface* that enables richer interaction (e.g., specify flow to move an object, use depth to steer viewpoint, use segments to select and edit). This induces a self-reinforcing cycle: improved predictions yield cleaner extractions; integrated structures expand and refine the control surfaces; and these, in turn, enable even better predictions and unlock new structures. Through this bootstrapping process, we progress from pixels to physics to broad scene understanding, supporting increasingly precise, local manipulation in the spirit of the rich representations envisioned by Marr [15].

Plan of this Paper: §§ 2—4 explore each of the steps of the PSI cycle in detail, and demonstrate the capabilities they unlock. §5 highlights several proof-of-principle applications of the model to diverse downstream tasks. §6 contains a discussion relating PSI to a panoply of related topics in AI and cognitive science, and details some important open questions and next steps in the PSI framework.

2 Probabilistic Prediction: Richly Controllable World Models

We now turn to building Ψ, the controllable predictor at the heart of PSI. The basic idea is to (1) break world data into a set of local variables and then (2) learn to predict the conditional probability distribution of each such variable, with the flexibility to condition on *any* subset of these variables. This idea is, in other words, to learn a *probabilistic graphical model (PGM)* [16] of the variables characterizing the world. With such a PGM in hand, a wide variety of inference pathways become possible, enabling fine-grained control. In the case of video data, our main illustrative example here, the natural place to start for breaking the data up is to use spatiotemporal patches. In this case, therefore, Ψ begins as a probabilistic graphical model predicting the likelihood of observing any subset of pixel patches at specified locations within a video, conditioned on the observation of zero or more other such patches elsewhere.

While the desire for a full probabilistic model of the world is clear, it might not at first be obvious how

to obtain this from data in an efficient and scalable fashion. The main technical innovation of this section is showing how this can be done, using a technique we call Local Random-Access Sequence (LRAS) modeling.

In what follows, we first describe Ψ's probabilistic prediction objective more formally (Section 2.1), and then introduce the LRAS technical formulation that enables learning Ψ efficiently (Section 2.2). We then illustrate how (Section 2.3) a specific instantiation of Ψ, trained on natural video data, can be used to perform a wide variety of inference paths, corresponding to useful controllability goals. Finally, we show how the probabilistic formulation supports a natural estimate of the entropy in scenes, and an iterative entropy-reducing "uncertainty management" process (Section 2.4).

2.1 Defining the Probabilistic Relationships Between the Variables of the World

The Ψ model seeks to predict local distributions over plausible future world states as a function of previous conditioning. To make this statement formal, we start with a data format that is described by a finite set of pointer locations $\mathcal{P}$ and a finite vocabulary $\mathcal{V}$ of content values. A *datum* $\mathbf{X}$ is any partially-defined function from $\mathcal{P}$ to $\mathcal{V}$, i.e., $\mathbf{X} : S \to \mathcal{V}$ where $S \subset \mathcal{P}$. A *pointer-structured data distribution* $\mathcal{D}$ is a distribution over such data. A canonical example of this concept is RGB video clips, where the pointers are space-time locations $\mathcal{P} = \{(i, j, t) | i \in [0, H-1], j \in [0, W-1], t \in [0, T-1]\}$ that range over a two-dimensional discrete array and a time variable; the values $\mathcal{V}$ are 8-bit pixel intensities in the three standard color channels; and the distribution of valid data instances is described by the (very complicated) natural statistics of the real world. At one level, the pointer-structured data concept is somewhat vacuous – it's hard to think of real-world data of interest that *isn't* pointer-structured. However, since our ultimate goal is to create autoregressive models (§2.2), it will turn out to be a useful formalism to handle the fact that many data types (e.g., vision) do not have a natural language-like uni-directional ordering.
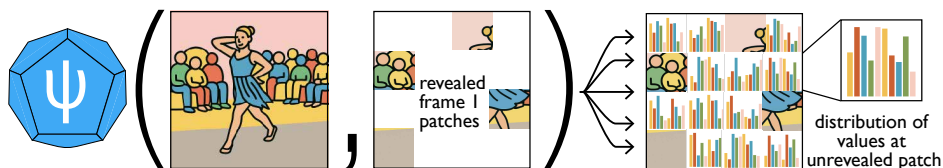
A key feature of pointer-structured data is that it is partially-defined in the pointer dimension, and potentially quite sparse. Partial definition expresses the concept of *conditioning*, in that a predictor given such data can be asked to return responses that are consistent with the inputs. For example, a valid datum might consist of a few patches of a single image, where natural "future" predictions complete the frame to a valid natural image. Another example that we will use throughout this work is a two-frame clip, i.e., data of the form $\mathbf{X} = f_0 \cup \{x_0, \ldots, x_k\}$, where f_0 is a full video frame and $\{x_0, \ldots, x_k\}$ are a selection of patches from a nearby subsequent frame f_1. Natural predictions of this type of data are completions of the second frame that are consistent with the appearance of objects in f_0 and object motions sparsely revealed by $\{x_0, \ldots, x_k\}$.

It is useful to think of the basic unit of pointer-structured data, the pointer-content pair $x = (p, v)$, as defining a single random variable in a probabilistic graphical model. The collection of all such random variables over all the pointer locations is the set of random variables whose relationships will be modeled. With these notions in mind, our core goal is to build a neural network for statistical completion of pointer-structured data. Formally, we are learning a function Ψ that takes as input any valid datum $\mathbf{X}$, together with a pointer $\mathbf{p}$ not already specified in (the domain of) $\mathbf{X}$, and returns the conditional marginal probability in the joint distribution $\mathcal{D}$ of observing value v at pointer $\mathbf{p}$, given conditioning $\mathbf{X}$. In symbols,

$$\Psi : (\mathbf{X}, \mathbf{p} \notin \text{dom}(\mathbf{X})) \longmapsto \{\Pr_{\mathcal{D}}[(\mathbf{p}, v) \mid \mathbf{X}] \text{ for } v \in \mathcal{V}\}. \tag{1}$$

In other terms, Ψ is a learned approximation to a probabilistic graphical model over the set of random variables corresponding to each pointer location.

To make this concept more concrete, it's useful to think about what it means in the context of the two-frame video data described above. In this case, Ψ is a function from a single video still frame, together with a few patches of information about the next frame, and which returns a spatial map of distributions of values present in the next frame at the unspecified patches:

In the above picture, the input to Ψ is a single frame, as well as the few patches fully specified; in the output panel on the right, the model outputs distributions of predicted values at each of the unspecified locations, conditioned on the inputs being present. Note that the sparse conditioning information about the second frame can, in this formulation, be of variable length. An empty set of conditioning patches corresponds to an *unconditional* prediction of the future frame, while as the size of the input patch set increases, further conditioning information is being provided, reducing the variability of the output distributions. With this idea in mind, it is convenient to define, for any datum $\mathbf{X}$ and subset of pointers $S \subset \mathcal{P} \setminus \text{dom}(\mathbf{X})$, the *map of distributions*:

$$\Psi_S[\mathbf{X}] \equiv \{\Psi(\mathbf{X}, \mathbf{p}) \text{ for all } \mathbf{p} \in S\}. \tag{2}$$

Since Ψ predicts distributions of plausible values at any pointer location, it can be sampled to generate instances. Given an initial datum $\mathbf{X}$, we generate values at unspecified locations $\hat{\mathcal{P}} = \mathcal{P} \setminus \text{dom}(\mathbf{X})$ by iteratively selecting and sampling subsets. At each step i, we choose locations $S_i \subseteq \hat{\mathcal{P}} \setminus \left(\bigcup_{k<i} S_k\right)$ that have not been generated yet, then sample from the conditional distributions:

$$\mathbf{X}_i \sim \Psi_{S_i}[\mathbf{X}_{i-1}] \quad \text{where} \quad \mathbf{X}_0 = \mathbf{X} \tag{3}$$

A major control knob for this process is the sampling order, described by the sequence of sets S_i used during generation. Two basic "pure" sampling-order strategies are available:

- **Sequential sampling**: Each subset contains exactly one location ($|S_i| = 1$), requiring n forward passes. This maximizes quality because each prediction attends to all previous patches.

- **Parallel sampling**: A single subset contains all locations ($S_1 = \hat{\mathcal{P}}$), enabling one-pass generation but assuming conditional independence among patches.

  Lying between sequential and parallel generation are a wide variety of hybrid strategies—including generating patches in semantic or spatial groups—presenting a fundamental trade-off between quality and efficiency (see §2.4). Once the sampling order is decided, the "usual" sampling techniques from language models (e.g., temperature control or top-k,top-p sampling) can also be used to control the diversity of predictions at each individual generation step.

  Throughout this paper, we use the notation:

$$\hat{\mathbf{X}} \sim \Psi[\mathbf{X}] \tag{4}$$

to denote fully sequential generation at all unspecified locations with random decoding order.
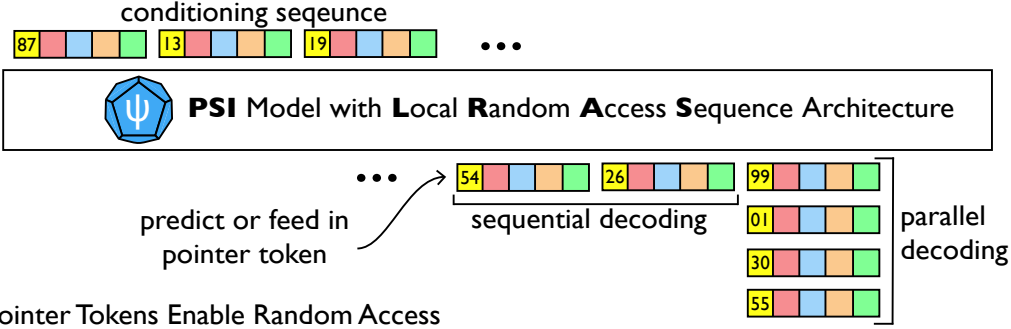
2.2   Local Random Access Sequence Modeling

A key question we have not yet addressed is: how exactly can we formulate a neural-network learning objective to efficiently learn Ψ as defined above. The key to this is the idea of the *Local Random-Access Sequence* (LRAS) model [17]. LRAS contains three key design decisions, which enable the implementation of the model described above:

Autoregressive Sequence Formulation. Because Ψ seeks to make predictions with partial conditioning of variable size, an architecture in which conditioning data are represented by fixed-size tensors is inappropriate. Our core insight is that this problem can be solved by reformulating Ψ as a sequence prediction problem (hence the "S" in its name), in which pointer values are simply woven into a sequence with one pointer $\mathbf{p}$ token before each value token v — so that each datum $\mathbf{X}$ is treated as a sequence $(\mathbf{p_0}, v_0, \mathbf{p_1}, v_1, \ldots)$. With this in mind, the LRAS formulation reduces to the standard autoregressive framework:

$$\Psi(\mathbf{X}, \mathbf{p}) = \Psi(\mathbf{X} \circ \mathbf{p}) = \Pr\left[v_k \mid \mathbf{p}_0, v_0, \mathbf{p}_1, v_1, \ldots \mathbf{p}_{k-1}, v_{k-1}, \mathbf{p}\right], \tag{5}$$

A. Autoregressive Sequence Formulation

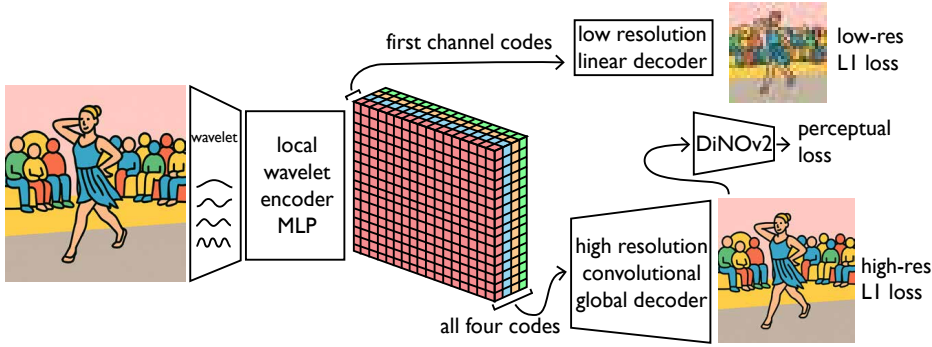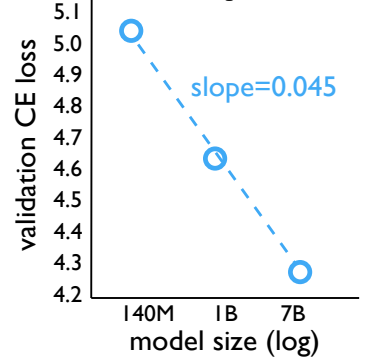Figure 2: **LRAS Architecture. (A)**: Hierarchical Local Quantizer (HLQ) encodes each local patch into a sequence of local codes. The first code decodes to a low-resolution version of the input, while the rest provide details to upsample the reconstruction to the original resolution. **(B)**: HLQ codes are transformed into the Pointer-Content representation to allow for random access to the patches when encoding and decoding the image causally. **(C)**: Modeling the serialized image representation with an autoregressive transformer. Pointer tokens can be either predicted or seeded while decoding. This allows for any amount of sequential and parallel prediction from the same KV cache. **(D)**: Scaling Laws illustrating our model obtains predictably lower loss when given more parameters.

and can be correspondingly trained with highly efficient sequence-to-sequence code. In this way, Ψ architecturally resembles the most successful world models in modern AI (LLMs). In fact, Ψ was designed to be optimized with the standard LLM pretraining procedure, utilizing the myriad of efficiency gains developed by that community to facilitate the production of ever larger models.

Pointer Tokens Enable Random Access.   Instead of the standard GPT architecture we introduce Local Random-Access Sequence (LRAS) modeling, which adapts the causal autoregressive modeling paradigm for higher-dimensional data without a clear serialization order. Traditional GPT-style transformers predict sequences in a preset, hard-coded order. While this is a natural fit for one-dimensional data such as language, it becomes an unnecessary, and potentially harmful inductive bias when modeling higher-dimensional data. Most autoregressive image modeling approaches simply accept this bias, while we introduce a new token type — the pointer — which allows us to serialize the data in arbitrary order.

Pointer tokens enable us to package random-access traversals over high-dimensional data structures (such as images) into one-dimensional sequences of tokens for efficient causal pretraining, by interleaving pointer tokens among the content tokens, which represent the actual data, as illustrated in Figure 2. During decoding, a *pointer* token i specifies the spatial index of the next patch to be emitted. The model may either *predict* a pointer (sampling the next region of interest) or *consume* an externally supplied pointer, allowing the user to guide the generation pattern.

In addition to freeing us from the raster-order generation bias, pointer tokens allow us to condition our model on any subset of the image, learning complex multidirectional conditioning relationships in the data. They also allow for partial patch conditioning, and patch regeneration during inference. All of this functionality can still be simply expressed as a causal autoregressive sequence of tokens, and thus can be modeled and optimized as a standard LLM.

Precise Controllability via Learned Local Quantizer. Most visual autoregressive models utilize popular off-the-shelf quantizers such as VQ-GAN [18], VQ-VAE [19], or the recent Cosmos tokenizer [9]. While such standard quantizers achieve strong compression ratios, they do not preserve locality of the patches within the token space, but rather encode the image in its entirety as a global code. While some locality is certainly present in the token representation, swapping any given token can modify the representations associated with patches on the other side of the image. While it is naturally more efficient to compress information globally, this comes at a cost of a less interpretable and less controllable latent (token) space.

Instead of compressing the whole frame into global codes, the LRAS architecture uses a Hierarchical Local Quantizer (HLQ): a convolutional autoencoder whose receptive field never crosses patch boundaries during encoding. Each patch is encoded into a sequence of four codes. The first code of each patch reconstructs a low-resolution preview, while the remaining codes add fine detail. No information from neighboring patches leaks into a code, preserving strict locality. This lets local downstream interventions—masking, overwriting, or re-ordering individual patches—behave predictably. Additionally, it makes the autoregressive modeling objective better aligned with natural-language modeling assumptions of token independence.

Key Training Details. We train an 80M-parameter HLQ on a combination of ImageNet and Open Images. We train a 7B-parameter Ψ model on a dataset of 3 million RGB video clips, consisting of about 1.4 trillion video tokens. We train Ψ with causal sequences of 2–4 frames, spanning up to 4 seconds of video. Mixed-precision training on 64 H100 GPUs at 65% MFU yields 490 TFLOPS/device ($\sim$31 PFLOPS total) sustained.

One critical "detail" of training is the use of the Warmup-Stable-Decay (**WSD**) learning rate schedule [20]. In WSD, the learning rate is **W**armed-up briefly at the beginning of training, then held at a **S**table level for a period at as high a rate as possible while avoiding model divergence (with the S-phase length scaling with model size); and then, just before the model is desired for use, linearly **D**ecayed to zero just slowly enough to avoid divergence. It has been shown that WSD achieves similar or better performance than cosine schedules of the same length. The use of WSD is particularly important here not because of superior performance, however, but because it enables continual training in the integration step of PSI (see §4).

Scaling Properties of LRAS. A critical advantage of the LRAS architecture is its predictable scaling behavior, inheriting the well-established scaling laws of language models while extending them to visual domains. We trained Ψ models from 100M to 7B parameters and observed consistent improvements in validation loss across three orders of magnitude (Fig. 2D). The reliable scaling without saturation even at 7B parameters suggests that further scaling would yield continued benefits, validating that LRAS successfully transfers the scaling properties of autoregressive language models to structured visual data.

LRAS Makes Probabilistic Graphical Models Tractable. The LRAS formulation transforms the challenge of learning probabilistic graphical models at scale. The key insight is that our sequences represent traversals through the PGM—each pointer-content pair corresponds to visiting and observing a node in the graph. By modeling these traversals autoregressively, we approximate the full joint distribution through conditional factorization.

The pointer mechanism ensures we can still query any conditional distribution $p(x_i|x_S)$ for arbitrary subsets S, but now through tractable sequential prediction rather than exponentially complex full inference. This recasts the seemingly intractable problem of learning a complete PGM over high-dimensional visual data as a standard GPT-style modeling problem.

Figure 3: **Unconditional Promptable Prediction.** Using the same unified model Ψ, single-frame unconditional prompting produces diverse plausible futures that reflect learned motion priors.

2.3 Diverse Inference Pathways Confer Rich Controllability

The LRAS architecture's flexible conditioning mechanism enables a rich spectrum of inference pathways through different prompting strategies. By varying which patches are provided as conditioning versus left for prediction, we can control Ψ to perform diverse visual tasks—all without any task-specific training. Figures 3 and 4 illustrate some key inference pathways that emerge from this unified framework.

Unconditional Prediction. The simplest inference pathway predicts a future frame f_1 given only the current frame f_0:

$$f_1 \sim \Psi[f_0] \tag{6}$$

This represents the model's prior over how scenes evolve when no additional constraints are provided. The model must infer all motion from the static frame—guessing velocities, accelerations, and future trajectories. This leads to high diversity in samples as the model explores the full distribution of plausible futures. Providing multiple conditioning frames $f_0, ..., f_{t-1}$ constrains the trajectory by revealing motion patterns, though uncertainty about future dynamics remains.

Patch-Conditional Prediction. Since Ψ treats all patches uniformly during training, we can condition generation on any subset of patches from the target frame. Let f_1^P denote a sparse set of patches from frame 1:

$$f_1^{\bar{P}} \sim \Psi[f_0, f_1^P] \tag{7}$$

This dramatically constrains the prediction space—often fewer than 10% of patches suffice to collapse generation to near-deterministic completion. The model leverages temporal factorization: appearance information comes from f_0 while motion cues come from the sparse patches in f_1^P.

Figure 4: Patch-Conditional Promptable Prediction. With sparse patches from the target frame as conditioning, Ψ collapses uncertainty to a constrained completion; replacing patches with synthetic edits yields counterfactual manipulations while maintaining physical plausibility — all with the same unified model.

Crucially, these conditioning patches need not be factual. By copying and displacing patches from f_0 to create hypothetical motion, we can prompt Ψ to generate counterfactual scenarios. The model treats these

synthetic patches as ground truth and synthesizes plausible completions—enabling precise control over object motion and scene dynamics.

Camera-Conditional Prediction. When camera pose information is available, we can insert it directly into the sequence as additional tokens. Let $\mathbf{C}_{0\to1} \in \mathrm{SE}(3)$ represent the relative camera transformation:

$$f_1 \ \sim \ \Psi\big[f_0, \ \mathbf{C}_{0\to1}\big] \tag{8}$$

This conditions generation on known camera motion, enabling novel view synthesis. The model must still resolve ambiguities about scene scale and camera intrinsics, and must hallucinate previously occluded regions, leading to multimodal predictions that capture this inherent uncertainty.

Camera Conditional Prediction

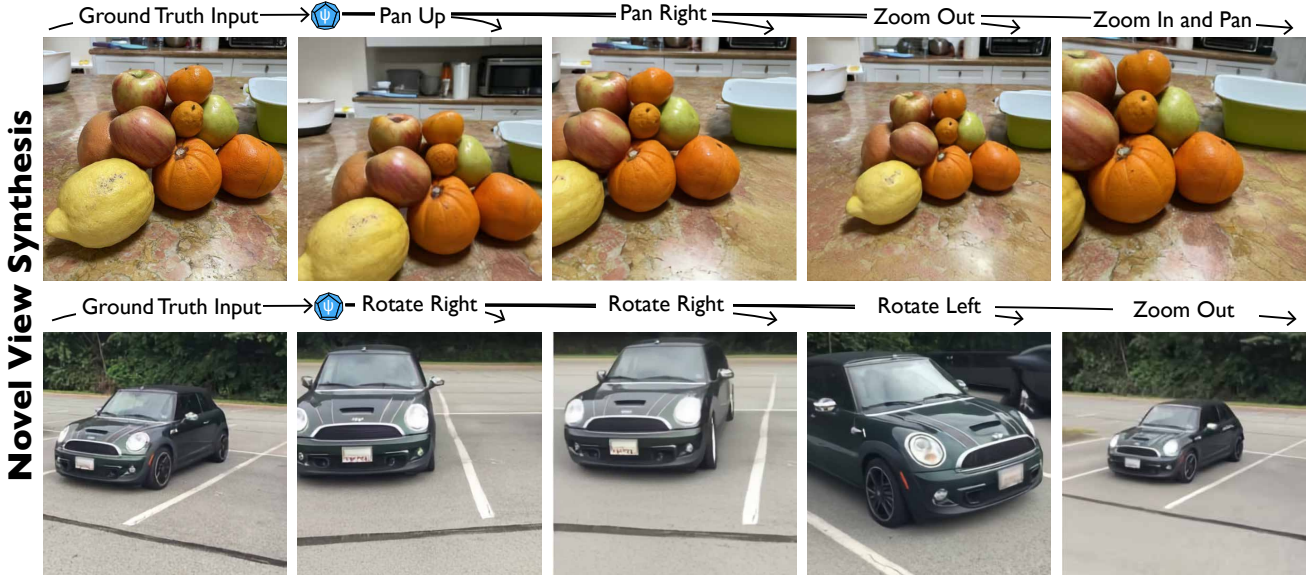**Figure 5: Camera-Conditional Prediction (Novel View Synthesis).** Given camera transformation parameters, Ψ synthesizes new viewpoints that preserve scene consistency while hallucinating occluded regions, using the same unified model.

Put differently, these prompting patterns correspond to different conditional distributions over the visual data: unconditional prediction yields $p(f_1 \mid f_0)$, where the model must infer all future dynamics from a single frame; providing factual patches from the target frame conditions on partial observations $p(f_1^{\bar{P}} \mid f_0, f_1^P)$, constraining the completion to be consistent with the revealed patches; substituting synthetic patches implements a causal intervention $p(f_1^{\bar{P}} \mid f_0, \mathrm{do}(f_1^P = \tilde{f}^P))$, forcing the model to generate completions consistent with the counterfactual evidence; and adding camera pose tokens conditions on additional observed variables $p(f_1 \mid f_0, \mathbf{C}_{0\to1})$. In every case, the tokens supplied in the prompt are treated as observed evidence, and the model predicts the remaining tokens conditioned on this evidence.

These inference pathways are not separate capabilities but different prompts to the same unified model. The pointer-based architecture allows arbitrary combinations: we can simultaneously provide camera poses, factual patches, and counterfactual patches as conditioning. Each additional constraint reduces the entropy of the prediction, guiding Ψ toward specific solutions within the vast space of plausible futures.

2.4   Entropy and Uncertainty Management

The pointer-based architecture of Ψ enables flexible control over the generation process through its ability to decode patches either sequentially or in parallel. Sequential generation, where patches are sampled one at a

time with each attending to all previous patches, provides maximal quality but requires multiple forward passes. Parallel generation enables single-pass inference but assumes conditional independence among patches. Between these extremes, we can generate patches in arbitrary groups, trading off quality for speed.

This flexibility becomes particularly powerful when combined with entropy analysis. By measuring the entropy of patch-wise logits during parallel prediction, we obtain a spatial map of the model's uncertainty:

$$\mathcal{H} = -\sum_{v \in \mathcal{V}} \Psi[f_0, ...] \log \Psi[f_0, ...] \tag{9}$$

This entropy map $\mathcal{H}$ reveals which regions of the scene are most uncertain given current conditioning. As we sequentially generate or reveal patches, each new piece of information constrains the remaining predictions, progressively reducing uncertainty across the scene.

Figure 6: Progressive Uncertainty Reduction Through Sequential Conditioning. We perform a series of **parallel** predictions at each step to measure the model's **certainty** during generation. We reveal the patches that yield the highest entropy at each step to optimally resolve the scene uncertainty. Each column demonstrates how additional conditioning patches (yellow squares) progressively constrain the model's belief about the state of the world, and improve the parallel prediction. **Note that sequential rollouts will produce coherent, sharp generations with any amount of conditioning patches revealed, at the cost of more compute.**

Figure 6 illustrates this process on a scene with a camel in motion. Initially, entropy concentrates on the animal's limbs and edges—regions with highest motion variability. As patches are revealed (yellow squares), the model updates its beliefs: first recognizing that the camel has moved, then resolving ambiguity about its new position. The rightmost column shows how parallel predictions at each stage become progressively more deterministic, converging from diverse samples to near-identical outputs as uncertainty collapses.

This unified view of sequential generation and uncertainty has important implications. For time-critical applications like robotic control, we can use entropy to adaptively choose between sequential and parallel generation—using fast parallel decoding when entropy is low, but switching to sequential generation in high-uncertainty regions. The entropy map also identifies which patches most efficiently reduce global uncertainty, enabling principled strategies for active perception and sparse conditioning.

The ability to quantify and manage uncertainty while flexibly controlling generation speed makes Ψ practical for diverse applications, from real-time robot planning where speed matters more than perfect quality, to high-fidelity video generation where sequential consistency is paramount.

3   Structure Extraction: Prompts as Causal Inference

Section 2 shows how to produce a learnable PGM for natural videos, where the variables are local spatiotemporal patches, leading to a powerful generative model. However, it is natural to want to transcend patch-level variables so as to be able to create world-model controls at higher levels of abstraction — e.g., motions and objects rather than RGB pixels. Such control surfaces naturally manifest as *intermediate (visual) structures* — that is, visual quantities that humans can estimate from raw RGB data but which in most typical computer vision contexts are thought of as "supervised labels" — e.g., optical flow, depth, object segmentation boundaries, and so on. If such quantities were readily available, they would be natural control conditioners — it would, for instance, be easier and more effective to create object motion hypotheticals by directly placing a patch of non-zero optical flow somewhere on the object rather than by the kind of naive pixel translation implemented in Section 2.

3.1   Intermediate Structures Are Zero-Shot Causal Prompts

But where should these intermediate quantities come from? Of course, it might be possible to use third-party supervised extractors to estimate them as needed. However, it turns out that a more conceptually satisfying and robustly performant approach is available. A true world model can be prompted to perform tasks it was not explicitly trained for. In Section 2 we explored various atomic conditioning prompts that allow us to control the generation of the model. Now we will show how we can compose several of those prompts to extract intermediate structures in a zero-shot fashion from Ψ. The basic concept is that:

$$
\begin{aligned}
\text{Properties of interest} \;&=\; \text{Intermediate structures} \\
&=\; \text{Structured prompts comparing factual and} \\
&\qquad \text{hypothetical/counterfactual predictions from } \Psi \\
&=\; \text{Causal inferences from } \Psi \\
&=\; \text{Generators of new types of tokens.}
\end{aligned}
$$

The first and second equalities above are essentially the idea advanced in an earlier work on Counterfactual World Modeling [13], but here we apply (and improve) this concept using the LRAS model. The third equality is essentially the main observation of the theory of probabilistic graphical models [21], of which Ψ is (as noted above) an instance. The last equality will be exploited in the subsequent section to create improved, more richly controllable, versions of Ψ.

This overall formulation also mirrors the paradigm shift from probing to prompting present in language models (and see 6.1). Rather than training task-specific probes for each visual property, causal prompts serve as the visual analog to language prompts—structured counterfactuals that query scene properties zero-shot, without supervision. Each extraction from Ψ is obtained by holding most of the scene fixed, perturbing only the factor of interest — in probabilistic graphical model terms, this is the equivalent of performing one or more do() operator(s) on the PGM encoded by Ψ — and measuring the model's response. The result is a unified mechanism for structure extraction that is steered entirely through prompt engineering.

Compared to the regression-based Counterfactual World Model (CWM) [13], which pioneered the process of zero-shot quantity extraction with counterfactual prompts, Ψ is a distributional generative model, and thus, offers two key advances. Firstly, the generated predictions are trained to match real-world distributions, not just their means, producing sharp predictions in response to hypothetical prompting, which makes it easier to precisely isolate the intermediate structure in question, for example, sharply segmenting the boundaries of an object. Secondly, Ψ is probabilistic, allowing for the resolution of ambiguous situations where prompts have multiple discrete and valid answers. This enables Ψ to capture complex definitions of intermediates such as objects, which can have fuzzy definitions. Repeatedly, we will find that being able to perform "crisp hallucinations" — that is, correct hypotheticals — is key for making effective causal inferences — that is, for good structure extraction.

What follows are three sections that illustrate causal inference programs to create three key intermediate visual structures: *optical flow*, *object segments*, and *depth* using Ψ. These results are also discussed in separate standalone papers exploring each topic in greater depth: Kim et al. [22] for optical flow (also see [23]), Venkatesh et al. [24] for object segments, and Lee et al. [17] for depth (which also illustrates novel view synthesis and complex object edits).

3.2 Optical Flow via Tracer Counterfactuals

Causal Inference Definition: When performing next frame prediction, the value of any given pixel in f_1 is "caused" by the appearance of pixels in f_0, which constrains the prediction. What translates the values between the two frames is the motion of the pixels — otherwise known as an optical flow field. Therefore, we can expose this flow field by counterfactually perturbing the pixel values of f_0 and observing the impact on f_1, since the former causes the latter. By marking a dot in f_0 and recording where that dot moves to in f_1, we reveal the causal structure of motion.

The key advance of Ψ over the regression-based Counterfactual World Model (CWM) [13] lies in its distributional generative nature. Where CWM produces blurry mean predictions, Ψ generates sharp, coherent samples. This distinction is crucial for optical flow, where correspondence can be inherently ambiguous—a texture patch might plausibly match multiple locations, or an occluded region might have several valid disocclusion patterns. In such cases, the world admits multiple sharp, discrete futures, not a single blurry average. The mean of plausible solutions is almost never a valid solution itself. While these sharp predictions would cleanly propagate counterfactual tracers through the scene if decoded, Ψ's distributional formulation offers something even more powerful: we can directly measure how perturbations affect the predicted distributions without ever decoding to RGB.

Following the Counterfactual World Modeling approach introduced by Bear et al. [13], we extract optical flow by measuring the effect of a slight appearance perturbation on the next frame prediction. Specifically, we add a small Gaussian bump $\boldsymbol{\delta}$ to patch i in frame f_0, creating $\tilde{f}_0^{(i)} = f_0 + \boldsymbol{\delta}\, \mathbf{1}_{\text{patch}=i}$. We then query Ψ with both the original and perturbed frames, conditioning on a random subset $\mathcal{R}$ of revealed patches from f_1. In CWM, the difference is computed as an L2 distance between mean predictions, while Ψ, with its distributional nature, allows us to capture a richer metric — the KL divergence between the predicted distributions. The patch whose distribution shifts most under perturbation identifies where the tracer reappears:

$$j^{\star}(i) \;=\; \underset{j \in \mathcal{P}_{f_1} \setminus \mathcal{R}}{\arg\max} \, \mathrm{KL}\Big(\Psi[\tilde{f}_0^{(i)}, f_1^{\mathcal{R}}], \Psi[f_0, f_1^{\mathcal{R}}] \Big) \tag{10}$$

The displacement vector $\mathbf{u}_i = j^{\star}(i) - i$ yields the optical flow estimate. This distributional approach naturally handles multimodality, producing diverse yet individually coherent flow predictions rather than attempting to average incompatible motion hypotheses.
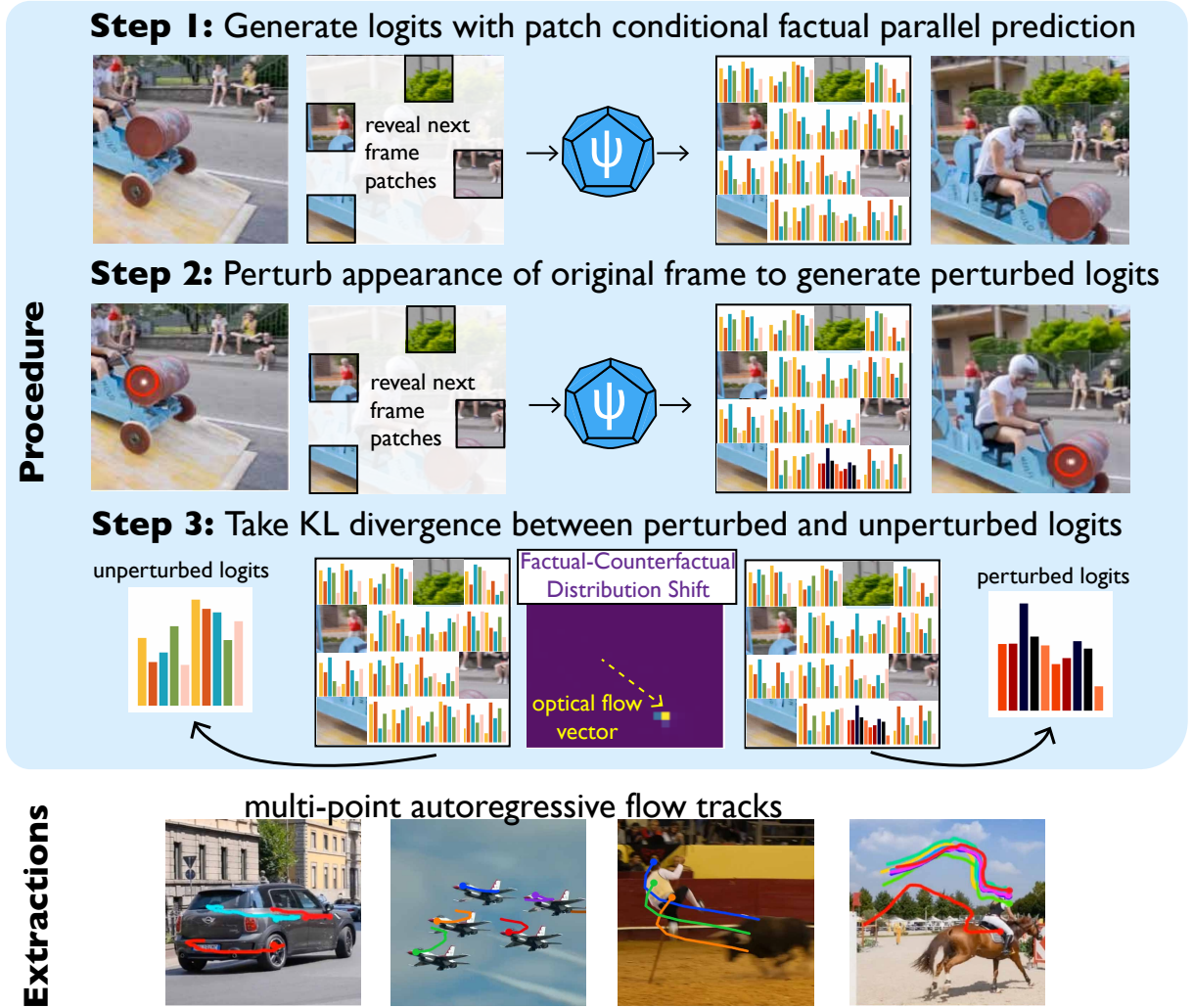
Procedure

Step 1: Generate logits with patch conditional factual parallel prediction

Step 2: Perturb appearance of original frame to generate perturbed logits

Step 3: Take KL divergence between perturbed and unperturbed logits

unperturbed logits   Factual-Counterfactual Distribution Shift   perturbed logits

optical flow vector

Extractions

multi-point autoregressive flow tracks

Figure 7: Optical Flow via KL Tracing. *Procedure (top):* **Step 1** — run patch-conditional *parallel* prediction to obtain per-patch logits; **Step 2** — add a tiny dot to f_0 to form a perturbed input and recompute logits; **Step 3** — take the KL divergence between perturbed and unperturbed predicted distributions to locate the correspondence in f_1, yielding a flow vector. *Extractions (bottom):* flow tracks across diverse scenes produced by repeating the probe at several locations.

	Method	DAVIS				Kubric			
		AJ ↑	AD ↓	$< \delta^x_{\text{avg}}$ ↑	OA ↑	AJ ↑	AD ↓	$< \delta^x_{\text{avg}}$ ↑	OA ↑
S	RAFT [25]	41.77	25.33	54.37	66.40	71.93	5.60	82.15	88.54
	SEA-RAFT [26]	43.41	20.18	58.69	66.34	**75.06**	6.54	**84.63**	**89.50**
W	Doduo [27]	23.34	<u>13.41</u>	<u>48.50</u>	47.91	54.98	<u>5.31</u>	72.20	73.56
U	SMURF [28]	<u>30.64</u>	27.28	44.18	59.15	**65.81**	6.81	<u>80.57</u>	**87.91**
	CWM [29, 30]	15.00	23.53	26.30	**76.63**	28.77	11.64	41.63	84.93
	PSI (ours)	**44.16**	**11.18**	**65.20**	<u>74.58</u>	<u>65.49</u>	**5.06**	**81.66**	<u>87.63</u>

Table 1: TAP-Vid First: quantitative results on DAVIS and Kubric. Tracking starts when a point first appears and continues to the video end, thus involving large frame gaps. Ψ outperforms two-frame baselines. "S" = supervised, "W" = weakly supervised, "U" = unsupervised.

15

Figure 7 illustrates this advantage. The top panel shows the KL-tracing procedure, computing divergence between clean and perturbed patch-conditioned predictions. The bottom panel demonstrates flow traces extracted from real-world dynamic scenes, tracking points across multiple frames using only the initial frame as conditioning. This zero-shot approach achieves state-of-the-art results on TAP-DAVIS and TAP-Kubric benchmarks (Table 1), surpassing both task-specific and world-model-based methods. For detailed analysis and quantification, see [22].

3.3   Object Segments via Motion Hypotheticals

Causal Inference Definition: When observing scene dynamics, the coherent motion of pixels across frames is "caused" by their membership in objects that move as unified entities. Groups of pixels move together because they belong to the same physical object—a latent structure that constrains their collective motion. Therefore, we can expose these object boundaries by hypothetically perturbing the motion of a small patch and observing which other pixels are compelled to move in concert, revealing the causal structure of object membership through their shared fate.

Following the approach pioneered by Bear et al. [13], we extract object segments by forcing Ψ to resolve motion hypotheticals. We copy a patch from f_0 and place it at a shifted position in f_1, creating the hypothetical prompt $f_1^{\Delta p} \sim \Psi[f_0, i \to i + \Delta p]$. Since the model understands that objects move coherently, it displaces all pixels belonging to the same object to maintain physical plausibility. We then compute optical flow between f_0 and the hypothetical prediction using SEA-RAFT [26], with the induced motion field revealing object boundaries. This transforms implicit object understanding into explicit segment tokens without task-specific supervision.[1]

The critical advance of Ψ over CWM lies in its distributional modeling of both object motion and background synthesis. While CWM produces mean predictions that can blur object boundaries, Ψ generates sharp hypothetical motions—diverse yet individually coherent samples that capture the multimodal nature of object behavior. This is particularly important for non-rigid objects that can deform and move in complex ways—a person's limbs might swing, a flag might flutter, or a tree's branches might sway. By sampling multiple hypothetical motions with different seeds, we can explore the space of plausible object behaviors and determine pixel membership through their consistent co-movement across samples. When an object moves hypothetically, Ψ must synthesize the previously hidden background with sharp detail—a task with multiple valid solutions that the model represents through its predictive distribution rather than collapsing to an averaged blur. Moreover, Ψ's multimodal predictions naturally handle the inherent ambiguity in object definitions: are a teacup and its plate one object or several? Are a rider and bicycle separate entities or a functional unit? Different samples might segment the scene differently based on which motion hypothesis is realized, aligning with the fuzzy, context-dependent nature of object perception where the same scene admits multiple valid interpretations.

Figure 8 showcases motion hypothetical prompts and their resulting segmentations. The model's sharp predictions enable clean object isolation — note how it appropriately generates occluded portions and respects depth ordering. For dynamic scenes, extreme counterfactuals (e.g., lifting a car upward) work particularly well, as they violate typical motion priors and force the model to isolate the object's causal influence. Quantitative performance is described in Section 4.3. More sophisticated algorithms leveraging this causal principle are discussed in [24].

---

[1]The results of this section mirror a more detailed presentation in a standalone paper on the use of the PSI technique to perform object segmentation [24].
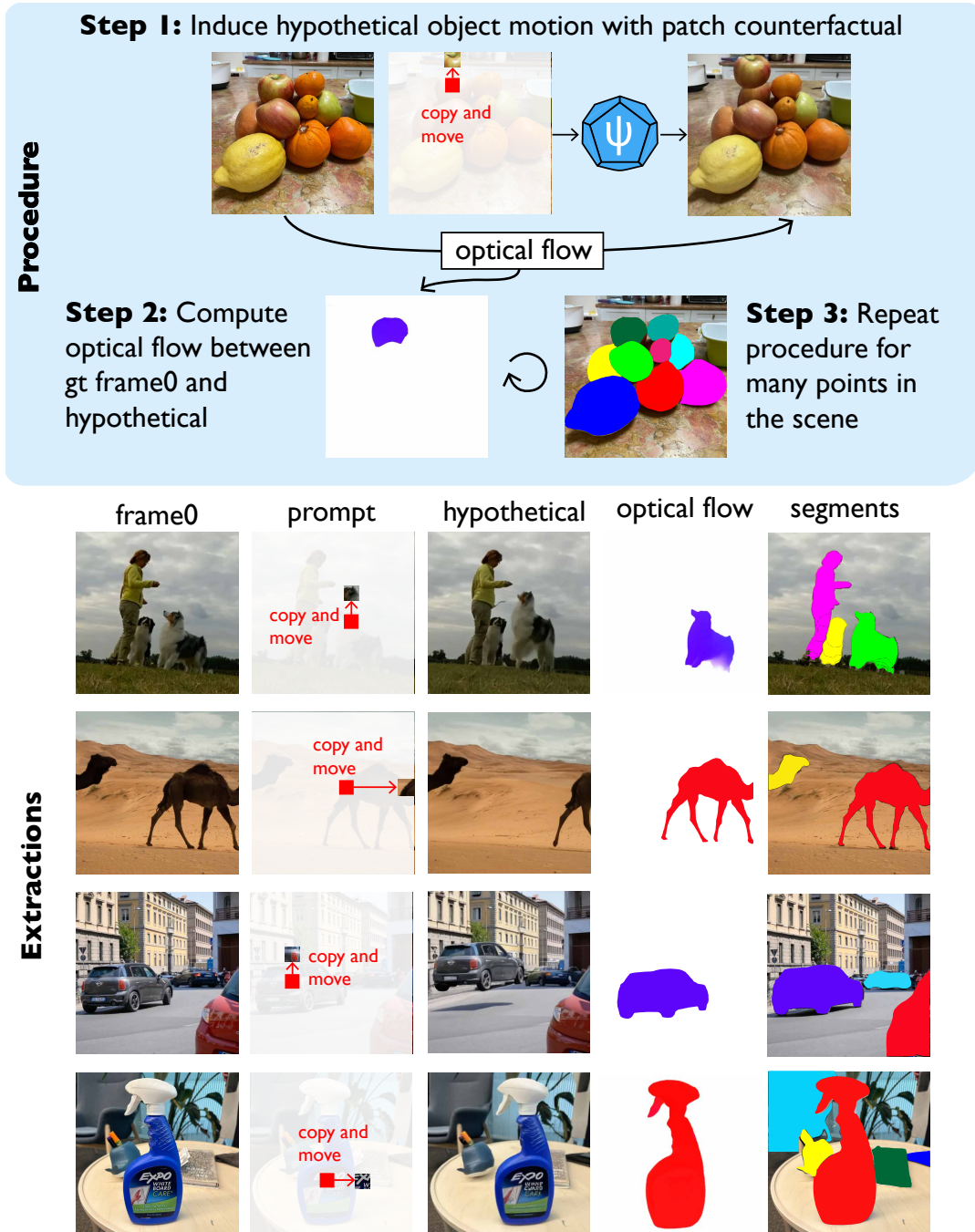
Figure 8: Object Segments from Motion Hypotheticals. *Procedure (top):* **Step 1** — induce object motion by copying a small patch in f_0 and moving it to a new location (counterfactual prompt); **Step 2** — generate the hypothetical frame and compute optical flow between f_0 and the hypothetical; **Step 3** — repeat over seed locations to cover multiple instances and convert coherent, induced motion into segments. *Extractions (bottom):* for each scene: frame f_0, prompt, hypothetical completion, induced optical flow, and the resulting instance segments.

3.4 Depth via Viewpoint Hypotheticals

Causal Inference Definition: When a camera translates through 3D space, the differential motion of pixels in the image plane is caused by their depth — the distance from the camera to the scene point. Near objects exhibit larger displacements than distant ones, a phenomenon known as motion parallax. Therefore, we can expose this depth structure by hypothetically perturbing the camera viewpoint and measuring the resulting parallax field, since depth is the latent cause that governs the magnitude of induced motion under viewpoint changes.

We extract depth by prompting Ψ to generate novel viewpoints. We append camera translation tokens $\mathbf{C}_\Delta \in \mathrm{SE}(3)$ representing a pure in-plane shift of magnitude b, yielding $f_1^{(\Delta)} \sim \Psi[f_0, \mathbf{C} \to \mathbf{C} + \mathbf{C}_\Delta]$. This creates an artificial stereo pair from which we compute optical flow using SEA-RAFT [26]. For each patch j, the horizontal disparity $d_j = |u_j|$ relates to depth through $z_j = b/d_j$, revealing the 3D structure without any depth supervision.[2]
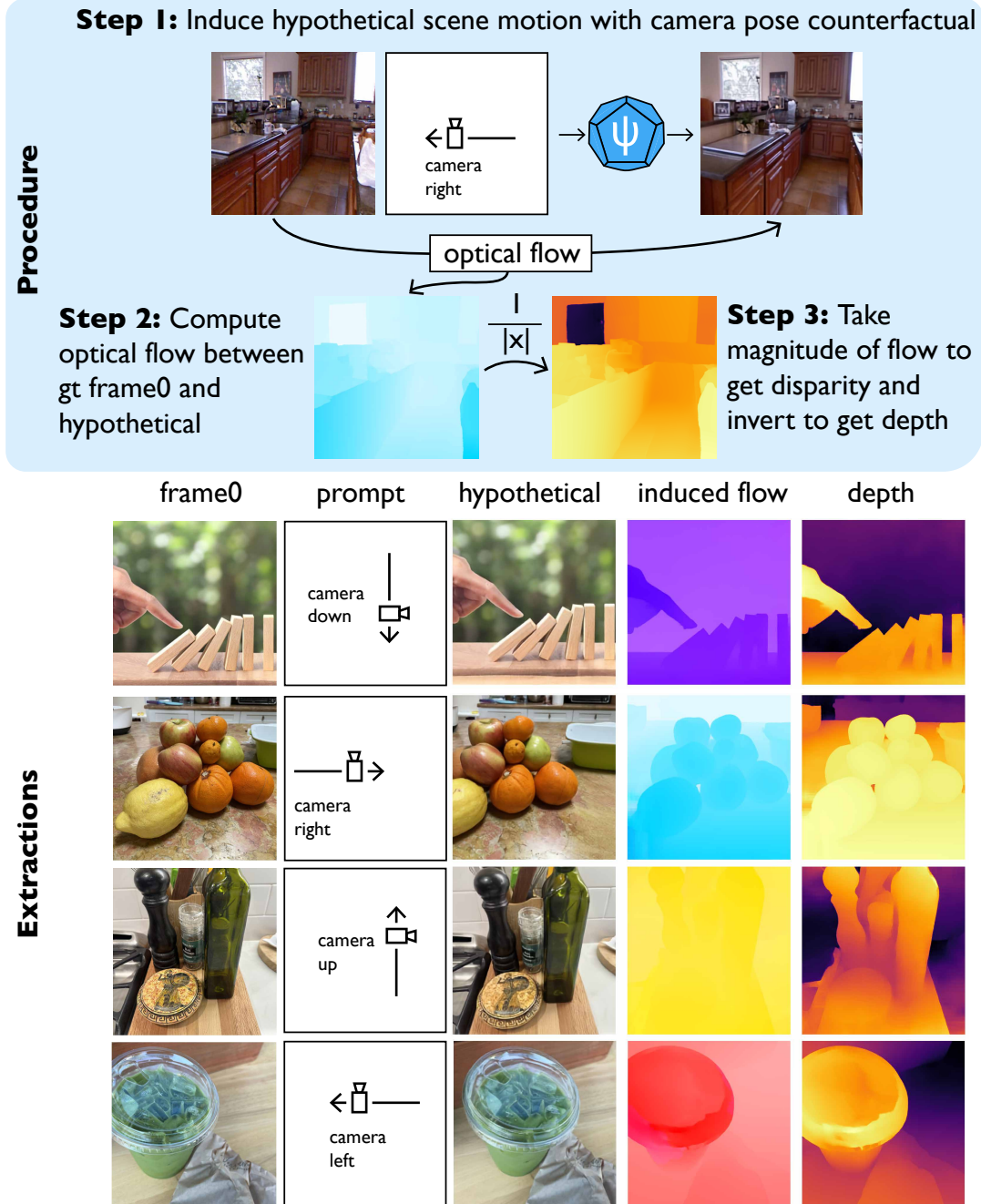

Figure 9: Depth from Viewpoint Hypotheticals. *Procedure (top):* **Step 1** — apply a camera-pose counterfactual (e.g., translate right/left/up/down) and generate the hypothetical view; **Step 2** — compute optical flow between f_0 and the hypothetical; **Step 3** — take the flow magnitude as disparity and invert to obtain (unscaled) depth. *Extractions (bottom):* per scene, we show the prompt, hypothetical, estimated flow, and resulting depth, illustrating consistent geometry across varied motions and baselines.

---

[2]The ideas of this section echo a more detailed presentation of results using the PSI technique to achieve 3D scene understanding, novel view synthesis, object editing, and unsupervised depth extraction in a standalone paper [17].

The advantage of Ψ's distributional modeling is particularly pronounced for depth extraction, where good hypotheticals — sharp novel views from camera motion — are essential for accurate parallax computation. While blurry predictions coming from regression models (like CWM) make precise stereo correspondences difficult, Ψ generates crisp, geometrically consistent views that enable clean disparity measurements. This sharpness is crucial at occlusion boundaries where Ψ must synthesize previously hidden regions with geometric accuracy. Moreover, Ψ's distributional nature elegantly handles depth ambiguities in textureless regions or through transparent surfaces, where multiple depth interpretations may be equally valid. Rather than collapsing to a single estimate or meaningless blur, Ψ represents multimodal depth distributions that capture inherent uncertainty. For dynamic scenes, Ψ's multimodal predictions are particularly valuable—sometimes the model predicts object motion along with camera motion, sometimes it does not. By averaging depth estimates across multiple samples, we can isolate the consistent depth signal from variable object dynamics, yielding robust depth maps even in moving scenes.

Note that this method builds upon optical flow as a fundamental extraction—the causal chain proceeds from depth to parallax to flow, highlighting how these intermediate representations compose. Figure 9 visualizes depth maps produced solely via viewpoint hypotheticals, demonstrating how Ψ's sharp predictions enable fine geometric detail. The model maintains consistency across different baseline translations, from subtle vertical shifts that freeze dynamic scenes to larger horizontal movements that reveal complex depth structure. Quantitative performance is described in Section 4.3.

3.5   Extracted Structures Define New Token Types

The structures extracted through causal inference are not merely outputs—they correspond to latent variables in the scene's probabilistic graphical model (PGM). Flow links past appearance to future position; depth mediates camera motion and parallax; segments capture object membership and coherent motion. Viewed this way, each structure is a natural token type: a discrete, local handle the model can read or write alongside RGB. As in language (characters to words to phrases), a multi-level visual vocabulary lets Ψ represent appearance, motion, geometry, and semantics within one sequence. Crucially, these tokens are bidirectional interfaces, serving as both conditioning inputs and prediction targets without task-specific heads or new vocabularies.

This perspective turns extraction into a bridge to integration. Once structures are tokenized, we can interleave them with RGB in a single autoregressive sequence to build a more expressive and powerful model. Section 4 operationalizes this bridge with a universal mechanism that mixes new token types into the sequence and continues training.

4   Integrating Structure: Ratcheting Prompting and Control

Having demonstrated how Ψ extracts fundamental visual structures through causal inference prompts, we now show how these structures can be integrated back into the model as new token types, dramatically expanding its prompting vocabulary. This integration transforms extracted intermediates from mere outputs into first-class citizens of the model's representational space, enabling richer and more precise interactions.

4.1   A Universal Mechanism for Structure Integration

Predictors that could directly observe and manipulate the new token types arising from structure extraction would be of great use. For example, if one built LRAS predictors combining flow tokens with the original RGB tokens in various combinations, it could unlock entirely new prediction "APIs", where the new tokens can be used *both* as inputs (e.g., conditioning signals) and outputs (e.g., prediction targets):

$$o \sim \Psi_1[f_0] \qquad o \sim \Psi_2[f_0, c] \qquad f_1 \sim \Psi_3[f_0, o].$$

In the above, Ψ_1 predicts the flow field o following f_0, learning the distribution of potential motions consistent with the input image f_0. Ψ_2 conditions on camera pose information c alongside f_0, disentangling global camera

motion from object motion in the predicted flow field. Ψ_3 uses both f_0 and o as conditioning to produce f_1—creating a powerful motion renderer that warps pixels according to the precise specifications of o, while synthesizing occluded regions.

What if one now wanted to integrate depth maps d in the same way? New predictors could capture these relationships:

$$d \sim \Psi_4[f_0] \qquad d \sim \Psi_5[f_0, c] \qquad f_1 \sim \Psi_6[f_0, d],$$
$$d \sim \Psi_7[f_0, o] \qquad d \sim \Psi_8[f_0, o, c] \qquad f_1 \sim \Psi_9[f_0, d, o].$$

Similar to the case with flow, Ψ_4, Ψ_5, Ψ_6 predict and condition on depth, while Ψ_7, Ψ_8, Ψ_9 combine flow and depth in various prediction and conditioning relationships. Each of these relationships is useful and complementary. However, the proliferation of useful predictor configurations needed to define these extended APIs is inconvenient. Must we train a zoo of specialized models to capture these manifold useful prediction problems? Many existing approaches would require architectural modifications for each new task—diffusion models need specialized conditioning mechanisms, regression models require new output heads. Even approaches that add extra channels or embedding heads suffer from parameter explosion and fail to scale computation with the increased information. Most critically, such architectures miss the key property we seek: establishing directional causal relationships between old and new tokens. A bidirectional transformer or encoder-decoder cannot specify whether flow causes RGB or RGB causes flow—but in our autoregressive formulation, the position of tokens in the sequence defines their causal ordering, making each new structure a proper node in the learned probabilistic graphical model.

It turns out that the design of LRAS enables a very simple but powerful unified solution to this problem. This three-step mechanism integrates any extracted structure into a unified model, requiring no architectural changes as new structures are added. This recipe transforms new structures into new tokens in the sequence, enabling seamless expansion of the model's capabilities through simple continual training.

Step 1. Tokenize the data and assign new pointers. For spatial intermediates such as optical flow and depth, we use the same HLQ architecture (Section 2.2) employed for RGB to produce 2D token grids that preserve locality and allow precise, patch-level control. For object- or scene-level intermediates such as segments, we adopt variable-length tokenizations that escape the *tyranny of the grid* [31], assigning as many tokens as needed per object rather than forcing a fixed patching. After tokenization, we add new *pointer addresses* for these structures by extending the pointer set $\mathcal{P}$ while keeping the *value vocabulary* $\mathcal{V}$ fixed and simply reuse the token indices for different modalities, differentiated by the pointers (Section 2.2). Thus, a previously RGB-only model can accept flow or depth tokens just by addressing them at new pointer locations—no new heads, no architecture changes, and, crucially, no vocabulary expansion that would cause parameter explosion. This design allows us to integrate an arbitrary number of new structures while keeping the model size stable, as each new modality only requires new pointer addresses rather than expanding the embedding matrices or output layers.

Step 2. Mix in the new tokens. New tokens are integrated by weaving them into the sequence that Ψ models—we take tokens from existing frames f_0 and f_1 and interleave the extracted intermediates between them, creating sequences like $[f_0, \text{camera}, \text{flow}, \text{depth}, \text{segments}, f_1]$ (as illustrated in Figure 10). This simple rearrangement defines all the predictors mentioned above (Ψ_1, Ψ_2, Ψ_3, ...) as different prompts of the same unified model, rather than requiring separate specialized architectures. The power of this approach lies in how tasks emerge from sequence structure itself: when RGB and flow tokens precede RGB tokens, we create a flow-conditioned frame predictor; when RGB tokens precede flow tokens, we create a motion predictor. The causal mask enables a single sequence to supervise multiple tasks in superposition—in a sequence $[f_0, \text{flow}, f_1]$, the first flow token sees the model as a motion predictor (conditioned on f_0), while the first f_1 token sees it as a flow-conditioned frame predictor, all within a single forward pass. This flexibility extends to arbitrary combinations: any subset of tokens can condition the prediction of any other subset [32], creating a vast space of possible prompts from a fixed set of integrated structures.

Step 3. Continue training. Because we employ the Warmup-Stable-Decay (WSD) training schedule, we can introduce new token types during the Stable phase and continue training from the last stable checkpoint seamlessly. The model integrates these new sequences into its training diet without catastrophic forgetting, maintaining performance on existing tasks while learning new capabilities. No architectural modifications or loss function changes are necessary—the model simply continues training with these augmented sequences.
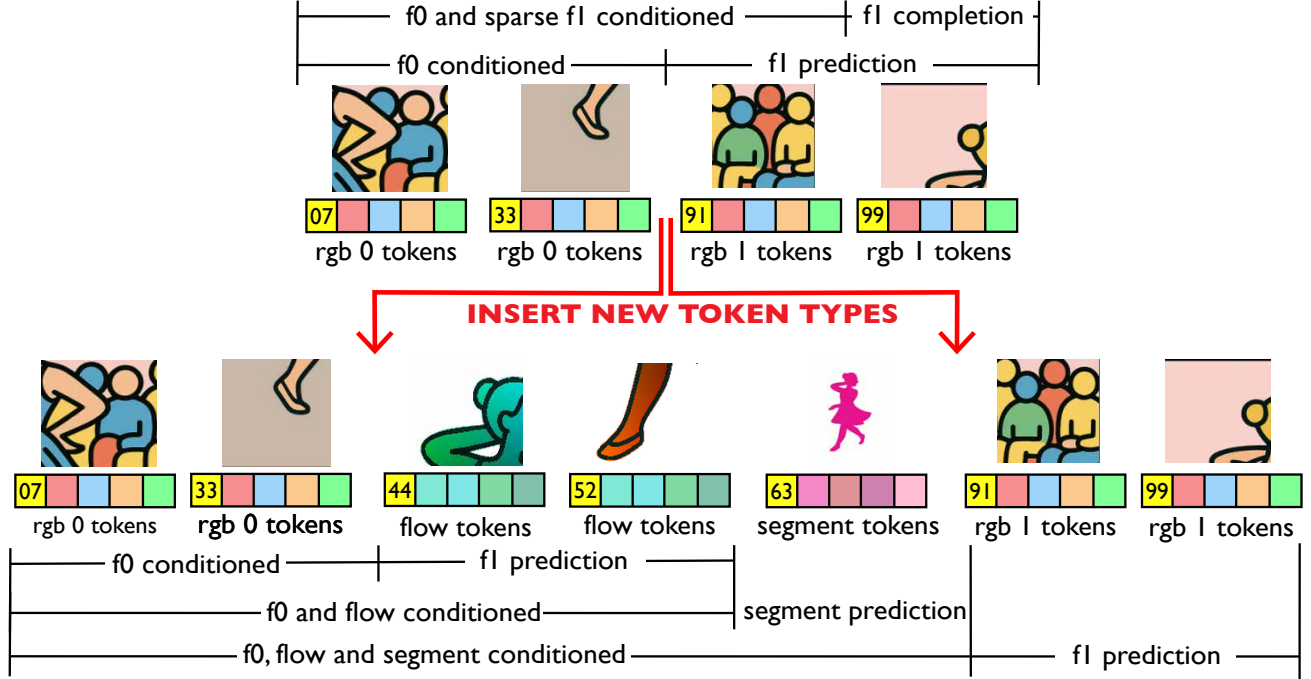

Figure 10: Mixing New Tokens Into Sequences (Integration Step 2). We integrate newly tokenized structures — e.g., flow, depth, and segments — by inserting their tokens between f_0 and f_1 in the same autoregressive sequence, following simple sequence-design principles. In this layout, the new tokens serve both as conditioning inputs and as prediction targets, decomposing the scene into core structures that steer the subsequent frame. The full training sequence therefore subsumes the "zoo" of predictors (e.g., motion prediction, flow-conditioned RGB, segment prediction) as masked subsets of a single sequence.

In the graphical-model view, the original pointer-indexed RGB (and known controls like camera pose) are the *observables* x, while the newly added token types (flow, depth, segments) are *latents* z. We introduce a *posterior-as-data* integration step: obtain approximate posterior samples $\tilde{z} \sim \hat{P}(z \mid x)$ via the counterfactual prompts of §3, interleave $(x, \tilde{z})$ in the same LRAS sequence, and continue maximum-likelihood training. This upgrades the model from $P_\theta(x)$ to the joint $P_\theta(x, z) = P_\theta(x) P_\theta(z \mid x)$. Because LRAS treats all pointer-addressed variables uniformly, the same parameters amortize a family of conditionals by toggling which tokens are observed versus predicted (e.g., $p_\theta(z \mid x)$, $p_\theta(x_S \mid x_{\bar{S}}, z_T)$, $p_\theta(z_T \mid x, z_{\bar{T}})$). Latents thus become first-class tokens in the joint; iterating with additional structures yields $P_\theta(x, z_1, \ldots, z_k)$ and a correspondingly richer set of controllable prompts.

In the subsequent sections we show that integration allows us to capture four key benefits:

- *Better control surfaces (§4.2):* By directly specifying intermediate structures — e.g., per-pixel motion — it is possible to specify desired manipulations much more precisely.

- *Better extractions (4.3):* The native prediction of intermediates as primary outputs combined with the precise control they enable produces higher-quality structures than post hoc extraction.

- *Access to further new structures (§4.4):* Intermediates themselves become sources for higher-order structures (e.g.,

motion entropy from flow predictions) that were previously inaccessible without the intermediate predictor.

- *Improved base prediction (§4.5):* factoring prediction through intermediates (e.g., first through predicting flow, and then predicting future RGB from past RGB and flow) decomposes hard problems into simpler steps, resolving different sources of uncertainty separately.

Together, these benefits create a virtuous cycle of continual improvement. Better predictions enable cleaner structure extraction, which when integrated enables even better predictions and unlocks previously inaccessible structures. Each cycle enriches Ψ's abilities: pixels lead to motion, motion reveals objects, objects enable affordances, building toward complete scene understanding. This recursive bootstrapping mirrors the hierarchical nature of visual understanding—from pixels to motion to affordances—with each integrated structure expanding the space of queryable properties and enabling second-order extractions that would be impossible from RGB alone.

Key Training Details. We validate our integration mechanism by integrating optical flow into a Ψ model initially trained only on RGB. We first train an HLQ tokenizer on flows extracted from our video dataset, then continue training Ψ on sequences augmented with flow tokens for an additional 0.5T tokens, resuming from the last Stable phase checkpoint of the RGB model training schedule. The model seamlessly learns to both predict and condition on flow without forgetting its RGB capabilities. The following sections explore how this single integration unlocks precise control, improved extractions, and better base predictions—demonstrating the power of the PSI framework.

4.2   Exposing Richer Control Surfaces

The integration of flow tokens allows us to exercise more precise control over the generations of Ψ. For example, we can overcome the fundamental limitations of RGB patch copying from Section 2.3, and instead specify motion through sparse flow vectors on an object as shown in Figure 11. Crucially, these flow specifications can be as sparse as needed—from a single vector to induce translation, to a handful of vectors placed anywhere on the object to specify more complex motions. Let $\mathcal{O}$ be the set of patch indices belonging to the object we wish to move, and $\mathbf{v}_i$ the desired motion vectors. The model propagates these sparse constraints into complete, physically plausible motion fields:

$$\mathbf{F}_{\mathsf{sparse}} = \{(\mathbf{p}_i, \mathbf{v}_i) : i \in \mathcal{O}\}, \quad f_1 \sim \Psi[f_0, \mathbf{F}_{\mathsf{sparse}}]$$

For rigid objects, sparse flows anywhere on the object generate coherent motion fields respecting rigidity; for non-rigid materials like fabric (Figure 11), the same sparse flows produce appropriate deformations based on material properties.

Camera control can also be made more precise with flow tokens. When using raw camera pose tokens, the model must infer camera intrinsics and scene scale—leading to a distribution over many plausible interpretations. With flow tokens, we can compute the exact pixel displacements for a given camera motion and directly specify them:

$$\mathbf{F}_{\mathbf{T}} = \mathsf{HomographyFlow}(\mathbf{T}, \mathbf{K}, \hat{\mathbf{D}}), \quad f_1 \sim \Psi[f_0, \mathbf{F}_{\mathbf{T}}]$$

where $\mathbf{T}$ is the transformation, $\mathbf{K}$ the known camera intrinsics, and $\hat{\mathbf{D}}$ the estimated depth.

We evaluate these capabilities on two complementary benchmarks. WildRGB-D tests novel view synthesis on challenging real-world scenes with large viewpoint changes—a task that requires accurate 3D understanding and realistic disocclusion synthesis. 3DEditBench [24] evaluates physically plausible object manipulation through complex 3D transformations like rotations and inter-object occlusions, measuring both visual quality (PSNR, LPIPS) and whether the edited object lands in the correct location (Edit Adherence). As shown in Table 2, Ψ achieves state-of-the-art performance on both tasks. For novel view synthesis, our flow-based camera control significantly outperforms specialized models. For object manipulation, PSI surpasses all diffusion-based editing methods by a large margin, achieving 22.73 PSNR and 0.797 Edit Adherence. These results validate that separating motion from appearance through intermediate tokens enables both more precise control and better generation quality. Further analysis of these capabilities appears in [17].
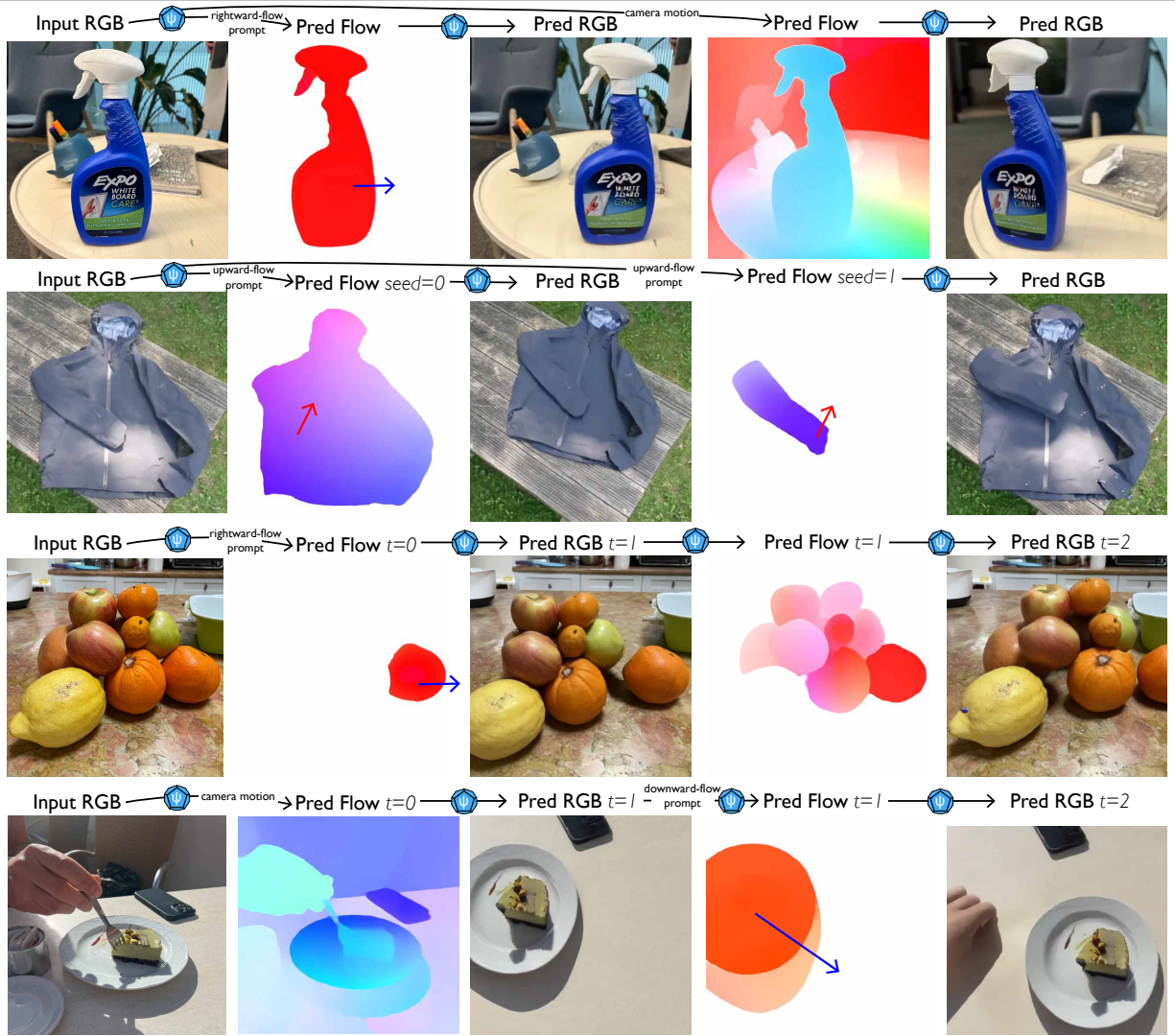
Figure 11: Improved Generation Control. By allowing PSI (Ψ) to accept flow tokens as conditioning, we expose a powerful new control surface to guide our generations. By strictly constraining the motion we wish our model to simulate, we significantly constrain the space of plausible outputs.

WildRGB-D: Novel View Synthesis

Model	PSNR ↑	SSIM ↑	LPIPS ↓
MotionCtrl	12.39	0.293	0.404
ZeroNVS	16.14	0.460	0.283
ViewCrafter	13.96	0.375	0.290
PSI$_{rgb}$	14.49	0.389	0.346
PSI	**18.27**	**0.566**	**0.177**

3DEditBench: Object Manipulation

Model	PSNR ↑	LPIPS ↓	EA ↑
DragAnything	15.13	0.443	0.517
Diffusion Handles	17.82	0.344	0.619
LightningDrag	19.52	0.184	0.722
PSI	**22.73**	**0.133**	**0.797**

Table 2: Quantitative results on novel view synthesis and object manipulation. Left: On WildRGB-D, PSI with flow-based camera control outperforms specialized novel view synthesis models. Right: On 3DEditBench [24], PSI achieves state-of-the-art performance for physically plausible object manipulation, significantly outperforming diffusion-based editing methods.

4.3   Improving Structure Extractions

The integration of new tokens fundamentally enhances Ψ's ability to extract structures in two distinct ways. First, the precise control surfaces discussed in the previous section enable more targeted causal inference prompts. Our original approach to object segmentation (Section 3.3) relied on copying RGB patches to induce motion—an approach that conflated appearance changes with actual movement. With flow tokens integrated, we can now provide pure motion hypotheticals: applying a single sparse flow vector to an object isolates its motion without introducing appearance artifacts. Second, by making optical flow itself a prediction target, we directly query for motion in its native representation. When the model predicts which pixels will move in response to our prompts, it outputs motion fields that inherently separate movement from lighting changes or shadows. This yields dramatically cleaner segmentations, as we're isolating exactly the property that defines objects—their tendency to move coherently as unified entities.

Figure 12 demonstrates how segmentation can fail in two ways without integration: either the flow predictor must compute flow between RGB frames (introducing errors), or the model moves objects along with their shadows, failing to isolate physical boundaries. Flow-based prompts solve both issues by encoding pure motion—a natural consequence of this is that we can isolate shadows as the difference between RGB and flow hypotheticals, since RGB captures all appearance changes while flow captures only physical motion.

Camera-conditioned flow generation similarly improves depth extraction by outputting predictions in their native motion representation. Since depth manifests as differential motion—closer objects exhibit larger apparent motion than distant ones during camera movement—flow tokens directly encode this parallax signal, yielding sharper and more precise depth boundaries.

Quantitatively, these improvements are substantial. As shown in Table 3, flow-based segmentation achieves 0.57 mIoU on SpelkeEntitySeg, significantly outperforming both RGB-based methods and specialized segmentation models. For depth extraction, PSI achieves state-of-the-art self-supervised performance across multiple benchmarks, with particularly strong results on dynamic scenes where motion cues are essential. The key insight is that integrated flow tokens enable direct prediction of motion-based structures without the indirection of RGB generation, yielding both better quality and computational efficiency. This validates the core principle: the right intermediate representation fundamentally changes what the model can extract.
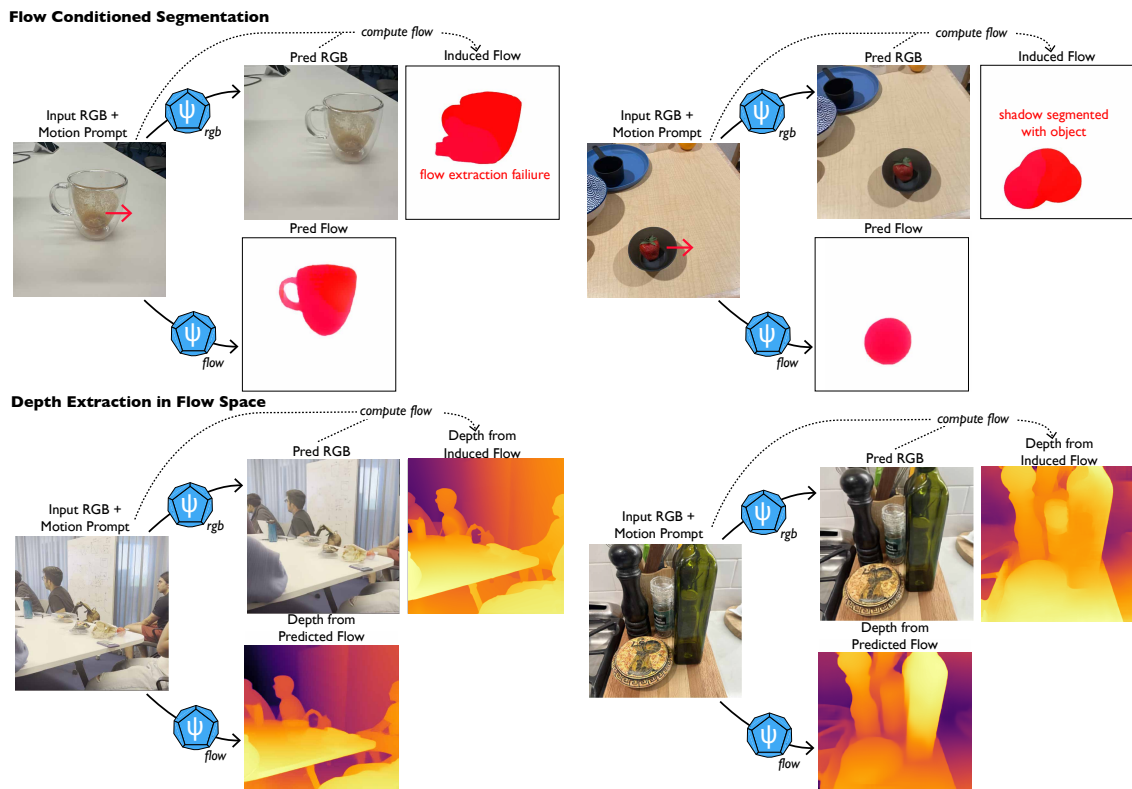
Figure 12: Improved Extractions Through Direct Intermediate Prediction. By including intermediate extraction tokens into the Ψ sequence modeling problem, we allow the model to be both conditioned on and predict the intermediates. Here we illustrate how flow can be used as conditioning to apply more precise motion edits to obtain cleaner object segments (top two rows), bypassing off-the-shelf flow extractor errors and ambiguous segments (such as objects). We also illustrate how directly predicting the optical flow field can improve the depth prediction by sidestepping the failure modes of off-the-shelf flow predictors and obtaining sharp, detailed depth estimates.

	SpelkeBench: Auto-Segmentation				**Monocular Depth Estimation** (δ_1)			
	SAM	DinoV2-G/14	$\textbf{PSI}_{rgb}$	**PSI**		MotionCtrl	SC-DepthV2	IndoorDepth	**PSI**
mIoU $\uparrow$	0.62	0.46	0.58	**0.65**	**NYUD** $\uparrow$	0.624	0.819	0.857	**0.873**
AR $\uparrow$	0.48	0.23	0.43	**0.54**	**BONN** $\uparrow$	0.798	0.800	0.827	**0.889**

Table 3: Improved structure extraction with integrated flow tokens. Auto-segmentation on SpelkeBench (left block: mIoU and AR) and self-supervised depth (δ_1) on NYUD and BONN (right block). PSI discovers physically coherent segments that outperform self-supervised methods and achieves state-of-the-art depth, particularly on dynamic scenes where traditional methods fail.

4.4 Bootstrapping Further New Tokens

The ability to model intermediate structures directly unlocks access to higher-order properties that emerge from the statistics of these intermediates themselves. A concrete example of which, we introduce probability of motion (P_{motion})—a property that emerges only after flow becomes part of Ψ's vocabulary. Unlike first-order extractions that reveal what is currently happening (like optical flow showing active motion), P_{motion} captures what is likely to happen: the probability that each patch will move in the immediate future, before any motion has occurred. As shown in Figure 13, this reveals striking patterns: in static scenes (top row), movable objects like cups and toys show higher motion probability than walls or floors; in kinematic scenes (middle row), objects with implied motion exhibit extreme probability concentrations; for articulated objects (bottom row), the model even captures movement hierarchies—a dog's head and tail show higher values than its torso. By querying the flow model's beliefs about future motion, we compute a statistical hypothetical—aggregating over all plausible motions that might occur:

$$P_{\text{motion}}(\mathbf{p}) = \sum_{\|\mathbf{v}\|>\epsilon} \Psi(\mathbf{F_p} = \mathbf{v} \mid f_0)$$

where we sum the probability mass assigned to all non-zero flow vectors $\mathbf{v}$ at each spatial position $\mathbf{p}$. This statistical aggregation captures the model's understanding of scene dynamics before any motion occurs. Unlike a single hypothetical that asks "what if this moved?", we're computing the expectation over all possible futures—revealing which pixels are poised for motion based on learned priors about how similar scenes typically evolve.
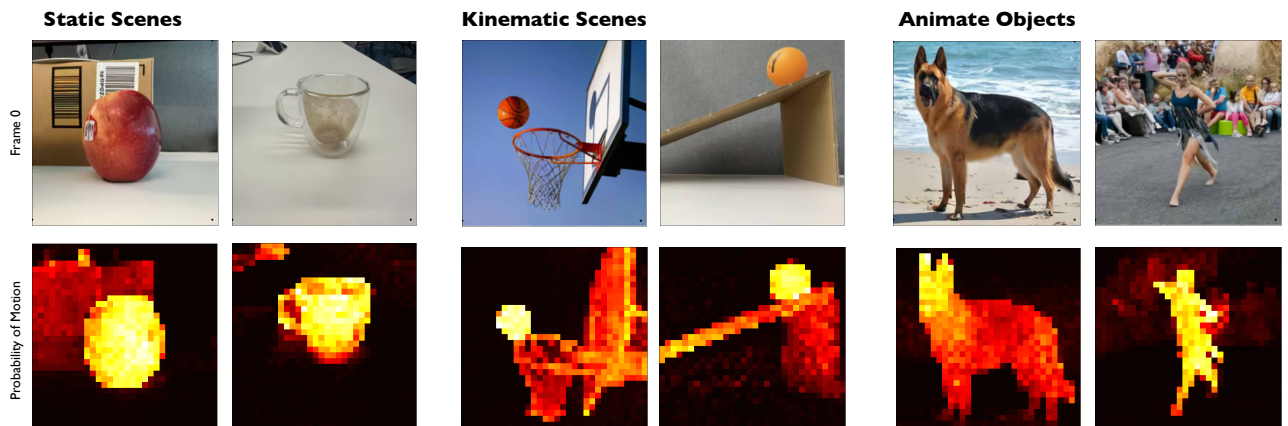
Figure 13: Probability of Motion. A parallel prediction is generated using Ψ_{flow} conditioned on the input image. Instead of sampling from the logits, the cumulative probability of all tokens corresponding to any amount of motion is reported. This yields a 2D heatmap of the objects in the image that are likely to move. Top row: static scenes where objects show higher motion probability than backgrounds and surfaces. Middle row: kinematic scenes with obvious implied motion, where active objects have extreme motion probability. Bottom row: animate objects exhibiting high motion probability, with fine-grained structure like the dog's head showing higher values than its body. Note that heatmaps are relative—static scenes have far lower absolute motion probability than dynamic ones.

Beyond binary motion probability, we can extract richer statistics from the flow model's predictions. Expected displacement ($\mathbb{E}[\mathbf{v}]$) captures not just whether motion might occur, but the most likely direction and magnitude at each spatial location:

$$\mathbb{E}[\mathbf{v}](\mathbf{p}) = \sum_{\mathbf{v}} \mathbf{v} \cdot \Psi(\mathbf{F_p} = \mathbf{v} \mid f_0)$$

where we compute the expectation over all possible flow vectors $\mathbf{v}$ weighted by their predicted probabilities. This reveals dominant motion patterns before they occur—a ball at rest shows expected displacement toward the ground due to gravity, a person mid-stride shows forward displacement, and rotating objects exhibit circular displacement fields. Unlike P_{motion} which captures motion likelihood, expected displacement encodes the model's beliefs about motion direction and magnitude, providing a vector field of anticipated dynamics that can inform planning and prediction tasks.

The progression from RGB to flow to motion probability illustrates a general principle: each integrated structure enables extraction of properties one level higher in the causal hierarchy. Flow tokens enable extraction of P_{motion} and refined object segments; integrated depth could enable extraction of surface normals, material properties, and 3D meshes; segments combined with flow could reveal functional relationships and object permanence. This recursive process creates an open-ended path where each new token type serves as both a useful representation and a stepping stone to richer abstractions. Through this bootstrapping, we can extract increasingly sophisticated scene representations—complete with geometry, dynamics, and affordances—from simple image inputs.

4.5   Upgrading the Underlying Predictor

One of the most important impacts of structure integration appears in Ψ's original task: RGB frame prediction. By decomposing complex video prediction into structured intermediate steps, we prevent a critical failure mode that plagues direct RGB models: motion collapse. When faced with ambiguous scenarios—a ball at the peak of its trajectory, a person mid-stride, a horse about to jump—models that predict RGB directly often hedge their bets by generating static frames.

Flow tokens force the model to commit to specific motion hypotheses. When predicting RGB directly, the space of all possible future frames—encompassing every combination of motion and appearance change—is so vast that models often assign most probability mass to the safe "no motion" prediction. In contrast, flow space focuses solely on plausible motions, a much more manageable distribution to model. This narrower scope prevents collapse to the static mode: the model can effectively distribute probability across actual motion patterns rather than defaulting to zero. The two-stage process—first predicting flow, then RGB conditioned on flow—transforms an intractable high-dimensional problem into two simpler sequential problems.

This factorization yields concrete improvements. Scenes that contain clear motion cues but produce static outputs in direct prediction models generate appropriate dynamics when motion is explicitly resolved first. Walking humans produce walking motions, galloping horses continue their gait, thrown objects follow ballistic trajectories. Rather than collapsing to static predictions, the model generates diverse samples that each commit to specific motion hypotheses. This captures uncertainty properly—through variation across different samples rather than through averaging motion directions to zero within individual predictions.

Quantitatively, PSI with flow integration outperforms both its RGB-only variant and the COSMOS baseline on VID across multiple datasets (Table 4). Notably, while COSMOS—trained without conditioning on text prompts—struggles to infer scene dynamics from visual information alone, PSI leverages its flow intermediate

to capture motion patterns directly from perceptual data. This demonstrates a fundamental advantage of our approach: by giving the model the vocabulary to reason explicitly about motion, we enable it to extract and utilize dynamic information that remains inaccessible to models limited to pixel-level representations.
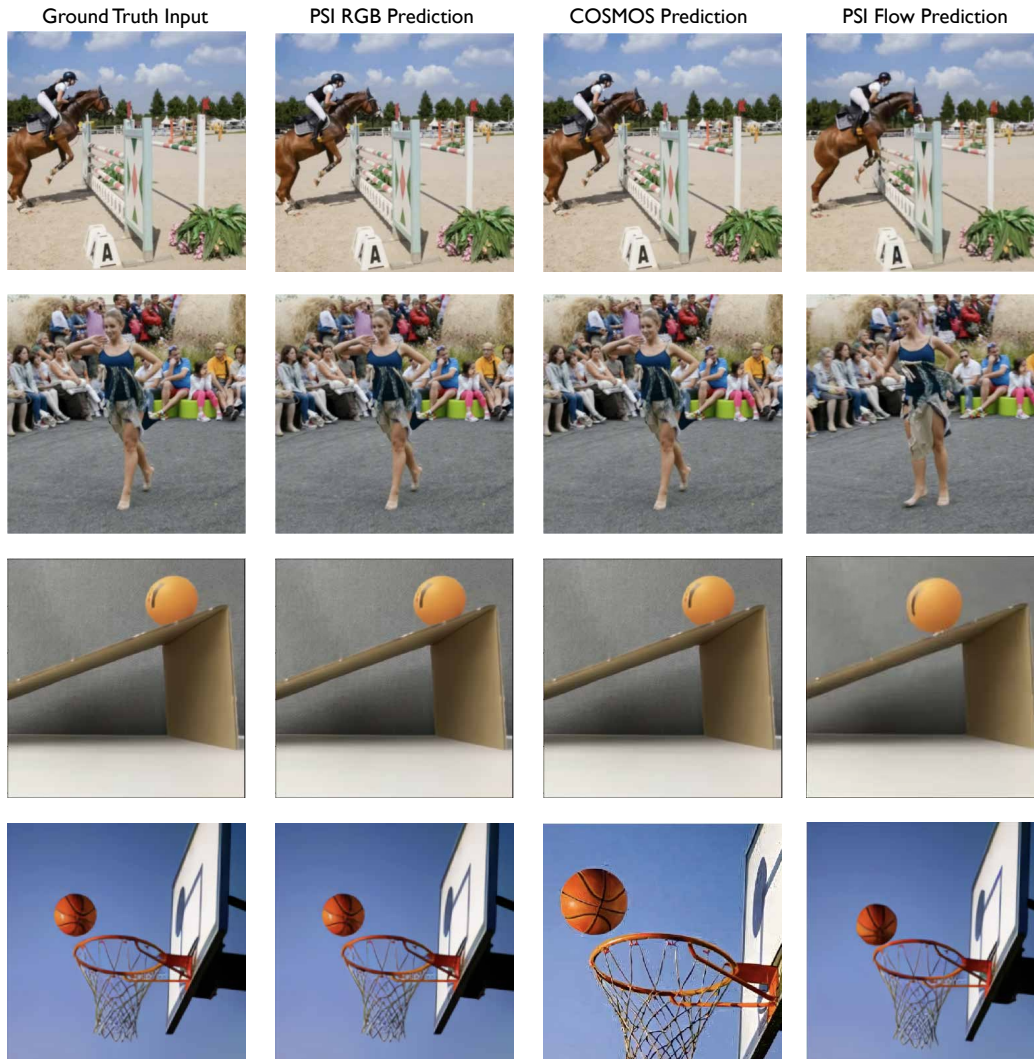

Figure 14: Unconditional Prediction with Flow Intermediate. Single-frame predictions from different models. Columns show: (1) input Frame 0, (2) PSI RGB-only prediction, (3) COSMOS 4B baseline, (4) PSI with flow intermediate, (5) ground truth Frame 1. Both RGB-only models exhibit motion collapse—generating static frames despite obvious motion cues. With flow integration, Ψ first predicts a motion field from Frame 0, then conditions RGB generation on this flow, successfully capturing dynamic scenes that stump direct prediction approaches.

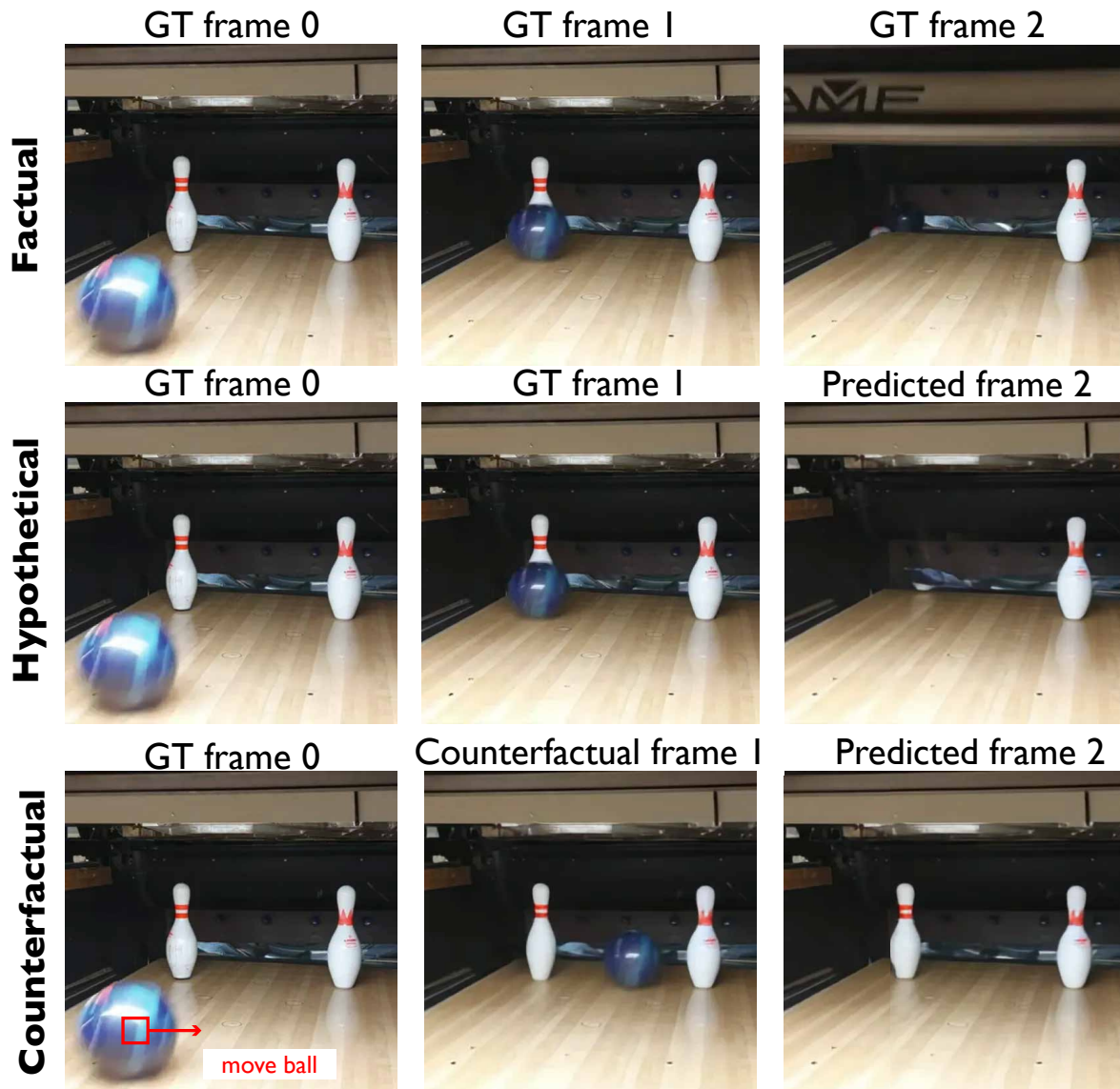| Model	DAVIS VID $\downarrow$	YouTube VID $\downarrow$
COSMOS AR 4B	210	340
PSI 7B (RGB Only)	223	307
PSI 7B (With Flow)	**198**	**173**

Table 4: Video prediction quality with and without flow integration. We evaluate single-frame video prediction across three diverse datasets: DAVIS [33] and YouTube—real-world videos containing interesting motion patterns. For each benchmark, models are conditioned on a single frame and asked to predict the frame 500 ms in the future.

5   Example Applications

Having described the PSI cycle in detail across the previous three sections (§2–4), we now showcase practical applications of the model to existing problems in computer vision and physical reasoning. Our goal is illustrative rather than exhaustive: to demonstrate how a single prompting interface enables factual rollouts, counterfactual analysis, and physically consistent manipulation across diverse tasks.

5.1   Physical Video Editing with PSI

PSI as a physical prediction engine for video editing. By conditioning on a small set of context frames, Ψ rolls out futures that respect contact, momentum transfer, and gravity; because the context is editable, localized pixel-level interventions (e.g., translating an object or toggling contact) act as low-level controls that deterministically recompute downstream effects—enabling factual, hypothetical, and counterfactual generation. In our bowling example, conditioning on two ground-truth frames yields a factual prediction that closely matches the true next frame—the ball strikes the pin and the pin topples—demonstrating that Ψ captures contact physics, while the same mechanism supports counterfactual edits that redirect the ball and prevent the pins from falling.

Intervene, then recompute the future. With a localized edit (we translate the ball mid-trajectory in the last context frame), PSI produces a *counterfactual* where the ball misses, so no force is transferred and the pins
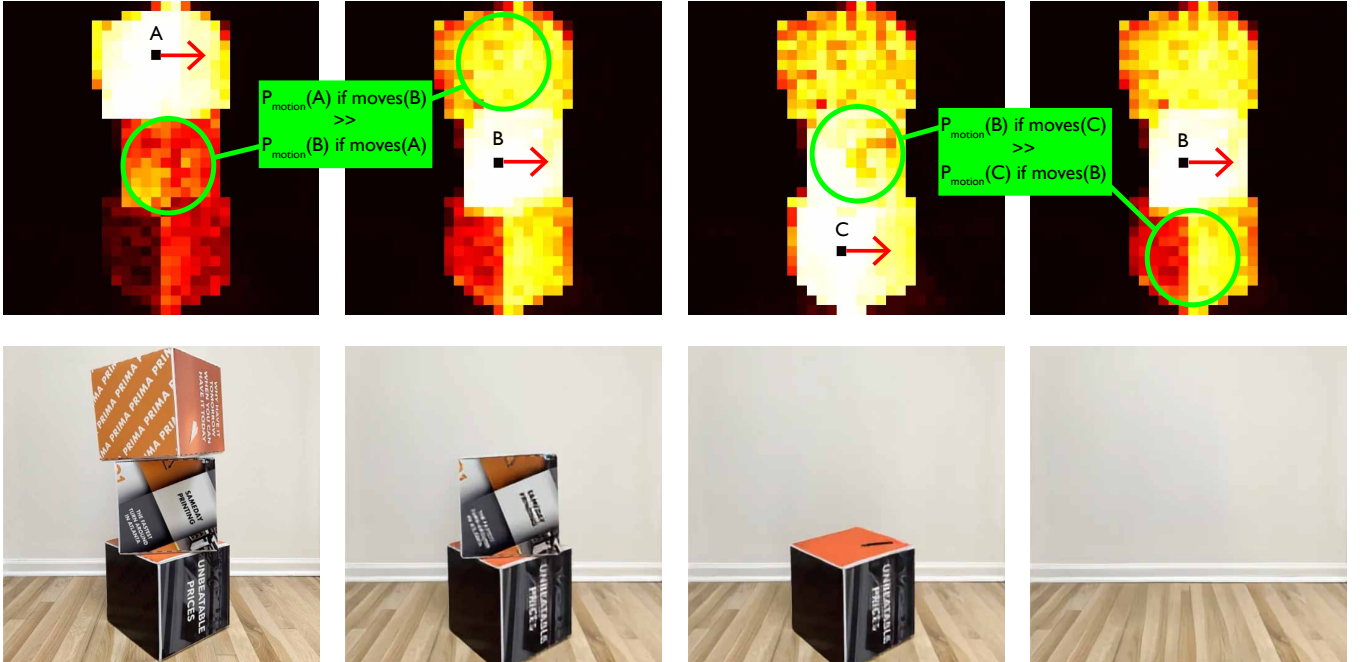
remain upright. In causal notation, this translation is a do-operation on the local (patch) variables controlling the ball's position at the edited time step, and the model propagates the consequences through the learned dependencies. This demonstrates controllable, physically consistent video manipulation: small edits to inputs yield coherent, downstream changes to outcomes.

5.2 Visual Jenga

The Visual Jenga task [34] requires identifying which object in a stacked scene can be safely removed without causing others to fall. We adopt the object segmentation approach from prior work but replace inpainting-based stability assessment with a physics-grounded solution using our flow-integrated model.

Bidirectional dependency testing. For each pair of objects (A, B) in the scene, we perform two complementary tests: (1) prompt object A with a small motion vector and measure the probability that object B moves in response, and (2) reverse the test by prompting B and measuring A's response probability. This yields an asymmetric dependency score $D(A \rightarrow B) \neq D(B \rightarrow A)$, capturing which object's motion more strongly influences the other.

Amodal completion through flow. Rather than using inpainting to remove objects—which can introduce unrealistic replacements or fail to respect physics—we leverage large-magnitude optical flow to achieve true amodal completion [17]. By applying flow vectors sufficient to translate an object entirely out of frame, we force the model to reveal occluded scene content while maintaining physical consistency. This approach ensures complete object removal without the hallucination artifacts common in inpainting methods.

Combining bidirectional dependency testing with flow-based amodal completion yields a robust solution to Visual Jenga: we identify the object with minimal outgoing dependencies (least likely to cause others to move when removed), then verify stability by applying amodal completion and checking that remaining objects maintain their positions. This physics-aware approach significantly outperforms inpainting baselines by respecting the causal structure of object interactions.
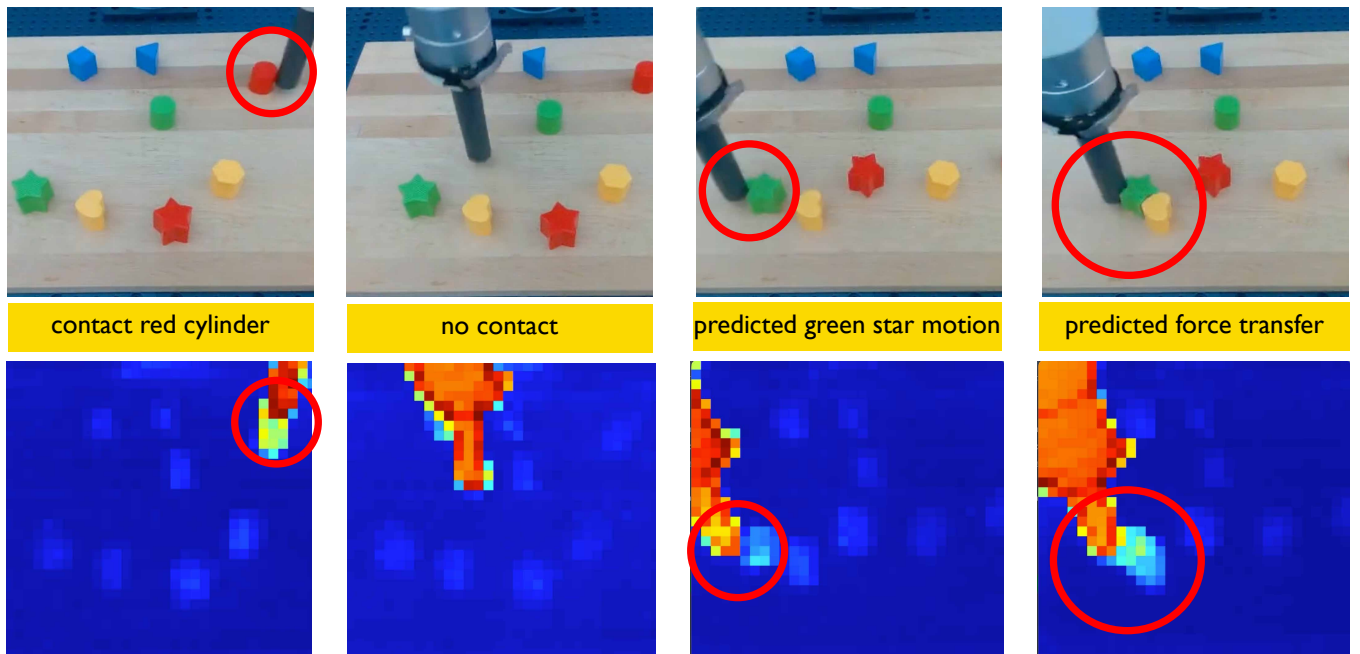
5.3 Robotic Motion Map

Robotic manipulation tasks require understanding not just what objects are present, but how they might move when interacted with. We demonstrate how PSI's probability of motion (P_{motion}) predictions enable robots to

anticipate dynamics before any motion occurs, simply by analyzing single-frame images.

Anticipating motion from static images. A key insight from our flow integration is that PSI learns to predict which objects are likely to move *before* any motion has occurred. By querying Ψ_1 with a single input frame f_0, we extract probability-of-motion maps that encode $P(\text{motion} \mid f_0)$—the likelihood that each patch will move given only the current static scene configuration. This capability emerges from the model's understanding of physical dynamics learned from video data.

Motion probability for robotic planning. For a robot manipulating objects on a tabletop, these motion-probability predictions from single frames provide crucial planning information. Objects with high motion probability are identified as manipulable, while low-probability regions indicate stable surfaces or immovable obstacles. The robot can query these probability fields to identify which objects are most likely to respond to manipulation, all from a single static observation. This anticipatory understanding—predicting what will move before applying any force—enables more intelligent grasp selection and manipulation strategies.

| contact red cylinder | no contact | predicted green star motion | predicted force transfer |

This single-frame motion prediction requires no interaction or motion observation to identify manipulable objects, generalizes to novel object configurations without task-specific training, and provides uncertainty estimates critical for safe manipulation. By anticipating motion from static observations, robots can plan more effectively before executing any actions.

6   Discussion

The PSI framework introduces several ideas. In summary:

1. By creating pointer-indexed local token sequences, and training an autoregressive model on these sequences in random order, we can create extremely scalable and highly controllable generative world models for complex real-world data, such as natural videos. These models act as full-fledged learned probabilistic graphical models of their data domains.

2. Useful quantities (also known as intermediate structures) are zero-shot causal inferences in the base predictor model, implemented via simple analysis programs that compare hypothetical (often counterfactual) predictions with ground truth observations.

3. The new "higher-order" token types emitted from step 2 can be integrated back into training data sequences using an utterly simple and generic token intermixing procedure, augmenting the base predictor with richer control surfaces and higher-fidelity prediction capacities.

Together, these ideas constitute a comprehensive framework for building powerful models of complex natural data domains. We illustrate these ideas in the domain of natural video.

The next sections link PSI to a variety of approaches in the literature, aiming to sketch a broader perspective on how these ideas interconnect.

6.1 Beyond Representation Learning: From Probes to Prompting to Integration

Representation learning is the idea of pretraining a network for one proxy objective, and then using the features arising from that optimization as a basis for supporting the learning of many other tasks [35]. Representation learning works on the principle that, if the proxy task is sufficiently broad, and the pretraining dataset sufficiently varied, the features needed to solve it will likely be useful for many other purposes. Initially, the proxy training objective was a supervised task (e.g. ImageNet categorization [36]), but the prohibitive cost and task-specific bias of large annotated corpora motivated a shift toward self-supervised objectives, including contrastive embeddings [37, 38], masked predictors [39, 40, 41], and autoencoders [42], which seek in various ways to align the learned model's latent variables with the environment's generative factors. Representation learning has proven very powerful and flexible, and applicable to pretrained models of many kinds, including modern generative AI [43].

The major limitation of representation learning is that it requires *probing* to answer any specific question – that is, the supervised training of a probe (often implemented as a linear or shallow MLP) on top of the pretrained representation. While this is less laborious than supervised pretraining of the base model, it is nonetheless dramatically less efficient and flexible than *prompting*, the paradigm associated with modern LLMs.

The LLM prompting paradigm has two key properties that distinguish it from representation learning: (a) one single model is sufficiently widely trained that it can itself answer most or all questions of interest, and (b) there is a generic "format" for posing all such questions directly to the model as zero-shot conditioners at inference time – as with natural language prompts. The shift from probing learned representations to zero-shot prompting exhibited in the transition from GPT-2 to GPT-3 is one of the singular achievements of human technology [44]. The flexibility of language prompts to formulate not just simple tasks (such as grammar checks [45]) but actually to contain cognitive strategies [46] and even define entire domains of intellectual endeavor, such as complex mathematics [47] or social agency [48], has remained a continual source of surprise and delight. Recent boundary-pushing work has taken this idea full cycle, using the outputs of inference-time constructs as a basis for creating synthetic training data for further base prediction-model improvements [14].

While prompting is a tremendously attractive concept, generalizing the prompting paradigm to the many domains where language-based descriptions are sparse or absent – such as video understanding and robotics – has proven challenging, both because it has remained difficult to extract quantities of interest in a zero-shot fashion from pretrained models, and because even conceptualizing how to formulate a "language" of prompts for such domains is unclear.

One major utility of the current work (but really starting with CWM [13]) is a route for beginning to shift non-linguistic domains toward a prompting strategy. The core insight was that a simple generic type of construction – counterfactual programs – could zero-shot extract most notions of interest, as long as the base predictor was properly formulated. The first two parts of PSI (probabilistic prediction and the structure extraction) can be thought of in analogy to the LLM case as (i) providing a generic approach to base predictors that, aside from their utility as highly controllable generative models, support (ii) zero-shot "prompts" that are highly effective. The third part of PSI, (iii) integration, exhibits a mechanism for using the zero-shot inference-time outputs as conditioners and targets to improve the base model, in analogy (but perhaps more generic) to the way synthetic data has recently begun bootstrapping LLM pretraining.

So why not just use text conditioning? The analogies we make to LLMs beg the obvious question: if language is so good, why not use it in your world models? Of course, that probably is a good idea, one

that many others have followed up [49, 50, 51], and which in the future will almost certainly continue to be of great import. Here, however, we have taken a different route. At one level this is because, as has become increasingly realized in recent literature from multiple domains [52, 53], solving many fine-grained and subtle problems of embodiment requires understanding that is not easily text-supervised given the nature of large-scale data available in the real world. Somehow, we will have to come to grips with fine-grained discovery and control of non-text data structures. However, there is a deeper reason why we have at least temporarily chosen to avoid text. While text conditioning might act as a strong signal for future prediction, it also gives the model a shortcut that is not always present in the real world. By forcing our model to infer scene transformations and motions purely from the input frame, we confront it with the same world-modeling prediction problem that intelligent life effectively solves, forcing it to learn a strong encoder function. It is possible that these non-textual solutions underlie the way cognition is learned in intelligent but pre-linguistic agents such as non-human primates and human infants [54, 55]. Ultimately, we do envision integrating language conditioning into PSI to provide an additional control surface–but only after first establishing sufficient visual intermediates (flow, depth, segments, meshes, and beyond) that can capture the full richness of scene dynamics without relying on the coarse abstractions that text necessarily imposes.

6.2   Related Work

Generative Models & Diffusion. Likelihood-based variational autoencoders [56] and adversarially trained GANs [57] first demonstrated scalable generative modeling, but encountered quality–diversity trade-offs for high-resolution imagery. Denoising diffusion models address these limitations through iterative refinement [58]; latent diffusion substantially reduces computational cost while preserving fidelity [59]. In discrete domains, autoregressive Transformers have shown very strong progress: ImageGPT [60] first extended next-token factor-ization to pixels, and contemporary large-scale language models [44] achieve strong likelihoods and emergent structure.
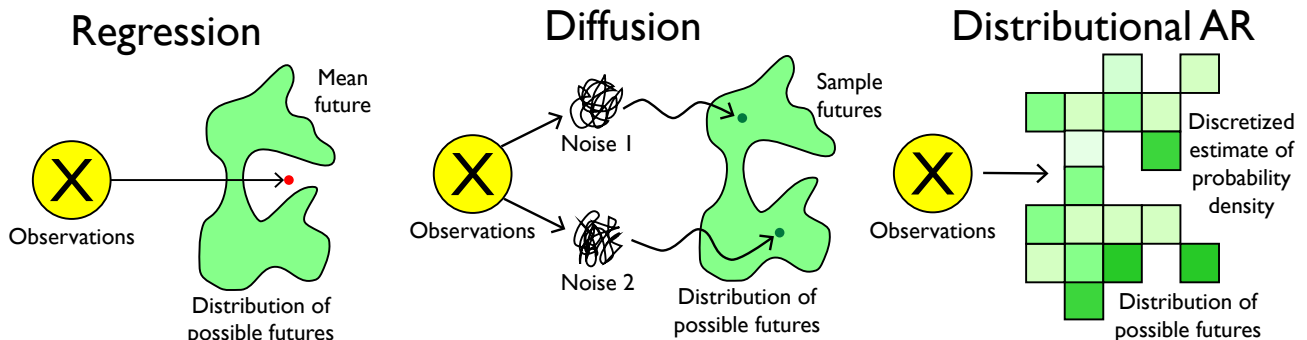

Figure 15: Regression, Diffusion, and Distributional Prediction. *Regression models* output the conditional mean of the future, which is often off-manifold – even though it minimizes distance to most possible futures. *Diffusion models* learn continuous paths that transform noise into realistic samples on the future-state manifold, but they do not model the entire probability density explicitly. *Distributional autoregressive models* (such as LLMs) output a full probability distribution over a discretized representation of the future, enabling explicit uncertainty estimates.

PSI's distributional LRAS-based predictor represents a choice to go all-in on a form of autoregression for generative modeling. By formulating the prediction problem fully distributionally (Fig. 15), unlike diffusion models which model one sample at a time, PSI's predictor makes statistical inferences that aggregate information across many potential future trajectories, such as entropy and motion maps, easily accessible. And despite their successes, both diffusion and existing autoregressive families typically condition generation on global embeddings or prompts. Localized conditioning has received comparatively limited attention but constitutes a focus of PSI. This choice has the key advantage of enabling precise spatial handles, which is useful both for precision of control at generation time, and for formulating effective hypotheticals for structure extraction.

Random Order Autoregressive Modeling. Several recent and concurrent works have explored random-order autoregressive modeling, though none combine the three properties that make LRAS uniquely suited for

controllable visual generation: random traversal of 2D data, strict token locality, and pointer-based addressing. Sigma-GPT [61] demonstrates that language models trained with random token permutations achieve comparable perplexity to fixed-order models while enabling flexible generation patterns. However, this work remains confined to 1D text sequences where the notion of spatial locality is less relevant. RandAR [62] brings random ordering to visual domains, training image models with arbitrary raster-scan patterns. Yet RandAR employs VQ-GAN tokenization, where each token encodes global information from a large receptive field—making local edits impossible without affecting distant image regions. XL-VAE [63] explores random orderings for image generation but focuses on hierarchical coarse-to-fine generation rather than maintaining strict spatial locality. The critical distinction of LRAS is that its combination of strict locality (via HLQ) and pointer-based random access enables the construction of a locally conditioned, queryable probabilistic graphical model.

From "Traditional" World Models to Data-Driven "World Models". A world model forecasts the consequences of actions. This concept has a long tradition within model-based reinforcement learning [64, 65] and model-predictive control [66]; Ha and Schmidhuber's *World Models* architecture [67]—a VAE coupled with a recurrent dynamics model—provided a simple and understandable toy example of this principle with modern neural networks. PlaNet [68], Dreamer [2], and Director [69] subsequently improved long-horizon planning by leveraging latent imagination in more sophisticated visual contexts. Recent lines of work have extended these ideas in a variety of directions [70, 71, 72, 73].

It might at first seem odd that we've discussed PSI as a framework for world modeling—after all, the inputs to Ψ are just data, so where are the actions whose consequences are to be forecasted? In a sense PSI (and the CWM concept that came before it) is a "poor man's world model", where expensive-to-obtain true action data is often proxied by cheap data patches that encode approximations to simple actions. That this proxy is effective is evidenced by the way that flow "tracers" and segment "motions" work well. The inferential form of Ψ formally treats data patches in precisely the same way that true action data would be, and reaps the reward of doing so because, as it turns out, training on raw data enables the underlying model to learn enough about the way the world works that it can competently perform hypotheticals (that is, enact do operators) without being hopelessly out-of-distribution.

The kind of "data-driven" world model represented by PSI has two advantages over traditional world modeling. First, it can learn most of what it needs to know about the world without the need for explicit action data, dramatically increasing its efficiency as a practical technological approach. Second, PSI's insistence on describing all predictions through the lens of potential "actions" forces a reckoning with the fact that no one observer is the only true agent in the environment; real things in the world outside the "agent" have causal status, and it is needlessly limiting to pretend otherwise. And of course, when real action data is available (as in the case of camera-conditioning discussed in §2.3), Ψ can seamlessly integrate it.

PSI vs Counterfactual World Modeling. Bear *et al.* introduce *Counterfactual World Modeling* (CWM) as a vision foundation-model framework that unifies a broad spectrum of visual tasks without recourse to task-specific supervision [13]. CWM rests on two interacting ingredients. First, a *structured masking* objective—implemented as *temporally-factored masking* in video—presents the predictor with an entire reference frame while masking almost all patches of the subsequent frame. This asymmetry forces the network to concentrate information about scene dynamics into a small set of visible tokens, effectively disentangling appearance from motion, and giving a strong interface for local control. Second, *counterfactual prompting* computes diverse mid-level representations by comparing the model's predictions on the actual input with its predictions on minimally perturbed ("counterfactual") versions of that input. Without any fine-tuning, a single pretrained CWM yields zero-shot estimates of keypoints, optical flow, occlusion boundaries, relative depth, and object segments. Subsequent work demonstrates that the same framework can be leveraged for physical reasoning. Venkatesh *et al.* [30] show that CWM's counterfactual queries extract structures—such as motion groups and contact events—that support state-of-the-art performance on the Physion dynamics benchmark, again without annotated training data. Stojanov *et al.* [23] show how CWM-style counterfactuals can be optimized to provide extremely effective optical flow extraction.

PSI solves two problems with CWM. First, CWM's underlying predictor was a regression-based masked autoencoder, leading to the "blurriness" associated with severe mode collapse, a limitation that affected many

downstream uses of CWM. PSI rectifies this by replacing the masked autoencoder with a fully distributional predictor capable of true "uncertainty management". Second, CWM provided no generic mechanism of integration — that is, using the outputs of structured extractions to improve the base world model (though the Opt-CWM framework in Stojanov *et al.* [23] shows one approach to this). PSI's reformulation in terms of sequence prediction, and the introduction of the sequence mixing concept for integration, rectifies this problem, enabling a fully virtuous cycle.

The cost of PSI over CWM is model size. PSI models have to solve a much harder problem than CWM — predicting a whole distribution rather than the mean tendency of that distribution — and achieving models that recover CWM functionality within the PSI framework (while solving many other problems along the way) requires substantially more parameters than in CWM. Thankfully, the amount of data for training does not need to scale significantly — bigger PSI models simply get more out of the data than smaller (or even bigger) CWM models can.

Learned Probabilistic and Bayesian Models. There have been a number of works seeking to learn PGMs from images and video data [74, 75, 76]. More recently, the probabilistic programming community has taken steps to integrate formal Bayesian perspectives on structure representation with learnable models, and applied these ideas to some computer vision problems [77, 78, 79]. Similar ideas motivate the addition of program-like structure to language models [80, 81]. Better understanding the connection between these works and PSI is also a subject for future investigation.

6.3   Predictions, Interventions, Hypotheticals & Counterfactuals

Pearl's ladder of causation [21] provides a framework in which to think about causal reasoning. In that framework, there is a clear distinction between predictions and interventions — that is, a difference between $p(y|X = x)$ and $p(y|\mathrm{do}(x))$ — relying on the well-definedness of the do operator for modifying (cutting links in) the underlying probabilistic graph of the system. Physically, applying the do operator corresponds to an exogenous process that changes the system's structural equation model (SEM) in a minimally-invasive fashion.

In the PSI setting, we are working with learned approximations to a very large PGM, where minimal modifications to the structural equations of the system are less well-defined and exogenous processes are inadmissible. We thus use a semantics of causation that is somewhat different from the classical Pearlian ladder of causation – following, in part, the suggestions of Tim Maudlin [82]. In our setting, *predictions* are the distributional outputs of Ψ when conditioned on ground truth observations, as well as any parallel-mode samples from these predictions; *intervention* is the act of obtaining predictions from Ψ on non ground truth (synthesized) conditioning; and *hypotheticals* are the sequential-model samples from such predictions. We do not make a bright-line distinction between $p(y|X = x)$ and $p(y|\mathrm{do}(x))$, but instead allow for a spectrum of intervention types expressed through different (non ground truth) input conditioning. The process of integration (§4), in which improved control surfaces are discovered, can be thought of as a way to ease the construction of "mini-surgeries" to the causal graph, converging toward the usual semantics of causal inference. A similar thought can be applied to methods of counterfactual optimization that achieve improved minimal interventions via a bootstrapping principle [23].

Though PSI does not admit exogenous processes (i.e. acts-of-god interventions outside the system dynamics), there is an equivalent in the PSI framework. As with any learned model, Ψ can be evaluated on unobserved conditioning inputs (that is, interventions in our sense) that are outside the training distribution of the model. Whether and where Ψ generalizes is an empirical question in each setting. Failure is detectable and the model can be refined, though not by changing the SEM explicitly — but rather by adding training data with a few of the right kinds of interventions observed. The hope is that the learned model will often generalize so that correct predictions can be achieved without having to have explicit examples of literally every version of intervention seen in some training observation.

One might wonder if causal inferences in this new semantics are as certain as they are under the usual ladder semantics. Indeed they are not – since they rely on building up "interventions" whose mechanisms are as certain as possible, but never quite have the absolute certainty that comes with being able to completely cut all the incoming connections to a given node. Even in the most certain case of causal analysis illustrated in this work – the tracer analysis yielding optical flow – it is always possible that the tracer will be interpreted by the

model as "schmutz on the camera lens" and not flow along with the object it is "on". Failure to measure the causal flow in this case will be confounded with not flowing at all. (Though this problem is not actually very significant in practice, as shown by decent results achieved with the method presented in §3.2, it can even be ameliorated in a principled fashion [23].) However – and in this regard we stand with Nancy Cartwright [83] – no *real-world* causal analysis can ever really be perfectly certain. Any physical manifestation of an intervention can only be implemented by systems relying on built-up empirical understandings resembling a learned model. For example, when a physician administers a treatment, even to know that she has picked up a syringe and successfully depressed the plunger to deliver the intramuscular injection — and not failed to do so in any number of byzantine ways – requires her sensorimotor system to have been trained (a key part of development) on the interpretation of visual, auditory, and haptic data, and their correlations with efferent copies of her own muscle discharges. There is causal uncertainty in this system of the same kind, and perhaps not lesser in magnitude, than that in PSI.

It is worth emphasizing the difference between hypotheticals and *counterfactuals*, since PSI does maintain the usual semantics of this distinction — and because the difference matters practically. Counterfactuals are a specific subset of causal inferences – those in which sufficient information can be provided *factually* so that the uncertainty about the effect of any hypotheticals are dramatically reduced, and perhaps eliminated entirely [84]. This reduction in uncertainty is how counterfactuals, when they can really be constructed, provide the ability to identify individual-level causal effects [85]. Counterfactuals thus enable stronger causal inferences as compared to "mere" hypotheticals [85].

Not all useful causal inference prompts to PSI are actually counterfactual – but some are. In the case of flow extraction (§3.2), the tracer hypothetical is truly counterfactual. This is because a decent amount of information about the second frame f_1 is provided to Ψ — in fact, enough information to reduce the uncertainty about object motion to zero, since it is only if the model is told what the actual motions of objects are from f_0 to f_1 that it can correctly predict where the hypothetical tracer should go.[3] For this reason, we are able to strongly identify the individual causal relationship between the f_1 point and its antecedent in f_0. Movable-object segmentation extraction from single frames is, by contrast, not actually counterfactual, because it merely involves comparisons between multiple hypotheticals, and the uncertainty in each individual prediction remains high throughout.

This contrast is quite instructive to analyze. The causal inference in flow is stronger than the causal inference in segments: indeed, movable-segment judgments from static images are subject to irreducible ambiguity, since, after all, an object, like a laptop on a table, that looks like it should be independently movable most of the time, can in any one instance actually be immovable (e.g. because someone surreptitiously glued the laptop to the table). Optical flow is a tighter concept with much less inherent ambiguity, precisely because you can "use much more ground truth" (all patches in both frames f_0 and f_1) when you estimate flow than you can when you estimate segments from a static image. This fundamental difference between the segments and flow concepts can be ultimately traced to the difference between mere hypotheticals and true counterfactuals.

It is also for this reason that flow is an excellent first intermediate structure on which to kick off a virtuous cycle of structural bootstrapping. Because it is a strong causal inference, using more information than subsequent downstream extractions, it can be estimated decently well by a Ψ predictor that has so far seen only raw RGB. This situation is an example of a more general phenomenon that is critical for continual learning to occur: the stronger the causal inference, the less powerful the predictor needed to estimate it well. Thus an early (not-yet-enriched) model can be sufficiently good to support extraction of strong causal inferences (such as flow), which can then allow a model enriched with the tokens emitted by that first round of extraction to become powerful enough as a predictor to better support extraction of weaker inferences (such as segmentation).

---

[3]Of course, if the provided f_1 patch happens to land where the tracer in f_0 would go, the hypothetical is quashed; these aberrant situations can be identified and removed by averaging over random choices of f_1 information provided; but this is also why we use *just enough* factual information to reduce object motion uncertainty to zero, but no more, so that as little quashing as possible happens. But the very possibility of this quashing is a sure sign that this is a *true* counterfactual.

6.4   Limitations and Key Open Problems

PSI has a number of limitations and leaves open a wide variety of future questions.

Integration of Global/Object-Centric Structures. In this work we show that spatially global structures (e.g. motion-derived object segments) can be extracted in a natural fashion using the PSI techniques. However, we only provide evidence of "closing the cycle" by integration gains for a local quantity (flow). We do not yet take advantage of global structures to create "object-centric" predictors. It is hoped that doing so would lead to both prediction performance and control precision gains. Ultimately, it will be useful to simultaneously integrate all the various intermediates we extract (e.g. flow, depth, segments).

Data domains beyond vision. The formulation of PSI is fairly general and largely independent of the specifics of the main example domain we present here, e.g. namely, natural video. It is thus natural to imagine applying PSI to a variety of other domains, such as neural response recordings, spatial transcriptomics data, geospatial, meteorological and cosmological data, and high-resolution spatio-temporal economics datasets.

Carrying out the first and third PSI steps in any such domain (creating an LRAS distributional predictor and integrating new token types via intermixing) seems comparatively straightforward. However, a key unknown for many domains will be how to take the second PSI step – namely, how to identify useful causal inferences. In computer vision, we rely on a clear set of intermediate structures that have been discovered over the years, and which have guided our selection of useful inferences in §3 of this paper. How will this happen in general, for domains where intermediate structures are less well-documented? This question motivates the next issue.

Automated identification of useful intermediate structures. To create the spectrum of structures extracted in §3, we created each causal probe by hand. A critical forward-going question for the PSI concept will be whether it is possible to generate useful causal probes in an automated fashion, without the need for hand-coded (that is, human-intelligence-guided) discovery. A successful approach to this problem would be a kind of high-level meta-principle of data-driven structure discovery, achieving a balance between the structure-skepticism of Sutton's Bitter Lesson [86] and the structure-optimism of developmental psychology [55] and the Probabilistic Programming approach [87].

Realizing this idea would probably involve some kind of Domain Specific Language (DSL) that could parameterize causal inference programs, possibly exploiting ideas from the literature on automated reasoning and inference discovery in probabilistic graphical models [88, 89] or language models [90]. It would also be natural if some principle of evolutionary utility could be used for causal program selection, coupled either with random or intrinsic curiosity-driven variation generation. It would also be natural to bolster this effort by deploying the principles for optimizing counterfactuals discussed in Stojanov *et al.* [23].

Learned Pointer Sequences. In this work we derive benefit from using random sequence orders in learning the Ψ model, and thus provide no training signal on the pointer tokens themselves. But because some generation sequences are better than others, as shown in (§2.4), it would be natural and perhaps very useful to use entropy information to bootstrap the learning of pointer token sequences.

Distillation of Extracted Structures. A limitation of the structure extraction paradigm of PSI (and CWM before it) is that hypothetical structure extraction is fairly slow. In recent work we have illustrated a simple approach to distillation that essentially treats the structures extracted as pseudo-labels for training fast single-purpose inference models [23]. More generally, it is possible to formulate the distillation problem as a sparse-to-dense prediction problem in the same sequential form as the existing PSI task, but using a modified sequence order (e.g. RGB0-RGB1-flow") that departs from the mix-in" order used in PSI integration (e.g. "RGB0-flow-RGB1"). Exploring these possibilities is a natural future work direction.

Where do semantic categories fit in? Many useful tasks that neural networks do involve explicit categorical concepts that have semantic value, such as high-level visual object categories like those that appear in the ImageNet ontology, or categorizing the actions of subjects in a video. Even when the networks are trained in a mostly self-supervised fashion (e.g. modern contrastive models [37, 38]), many of the loss functions for which such networks are trained produce "cluster-y" representations that enable simple probes (e.g. single linear layer) to recognize semantic category boundaries. Where are such concepts in PSI? It may be that PSI-learned encoders are effective at supporting category readouts in the same way as earlier self-supervised methods; but it is not obvious if they are as cluster-biased as contrastive self-supervision objectives are. Whether and to what

extent contrastive techniques should be integrated with PSI is an important question.

Long range predictions: how is memory to be integrated? While it is possible to scale PSI to learn decently long sequences – and thus understand/generate videos of at least a few seconds long, and probably substantially longer – the future of scaling both for PSI (and dedicated video generation models as well) will surely involve some kind of online-adaptive memory, akin to the hippocampal-cortical interaction [91]. Exactly how best to do this to achieve long-range generation and planning remains uncertain.

6.5   Does PSI have anything to do with human neuroscience or cognition?

Frankly, we don't know yet. However, this is an obvious target for future work, both because it is an interesting class of questions in itself, and because we as a team are, in other lines of work, often engaged in comparisons of AI models to human behaviors and brains. Our team's strategy for creating improved cognitive and neural models has always been to look for large capability gaps between existing AI approaches and what humans do, and then seek to fill these gaps with improved algorithmic ideas [92]. The principle behind this activity is what we have in other contexts called *contravariance*: that as models are increasingly strongly constrained to solve successively more challenging intelligence goals, they are increasingly likely to converge on mechanisms like those that evolution has created in people [93]. PSI can be thought of as an excursion into the *terra incognita* of novel algorithm spaces, relying on the contravariance principle to eventually bring us home onto solid scientific grounding. Time will tell if this is a sound strategy.

Acknowledgements

<div align="center">

***

S. D. G.

***

</div>

References

[1] Isaac Newton, I Bernard Cohen, and Anne Whitman. *The Principia: mathematical principles of natural philosophy*. Univ of California Press, 1999.

[2] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[3] David Ha and Jürgen Schmidhuber. World models. In *arXiv preprint arXiv:1803.10122*, 2018.

[4] David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David P. Reichert, Neil Rabinowitz, André Barreto, and Thomas Degris. The predictron: End-to-end learning and planning. In *34th International Conference on Machine Learning (ICML)*, pages 3191–3199, 2017.

[5] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[6] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023.

[7] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[8] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[9] Cosmos-Tokenizer Team. Cosmos tokenizer: A suite of image and video neural tokenizers. *arXiv preprint arXiv:2411.12223*, 2024.

[10] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[11] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[12] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[13] Daniel M. Bear, Kevin Feigelis, Honglin Chen, Wanhee Lee, Rahul Venkatesh, Klemen Kotar, Alex Durango, and Daniel L. K. Yamins. Unifying (machine) vision via counterfactual world modeling. *arXiv preprint arXiv:2306.01828*, 2023.

[14] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *CoRR*, abs/2212.10560, 2022.

[15] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.

[16] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[17] Wanhee Lee, Klemen Kotar, Rahul Mysore Venkatesh, Jared Watrous, Honglin Chen, Khai Loong Aw, and Daniel LK Yamins. 3d scene understanding through local random access sequence modeling. *arXiv preprint arXiv:2504.03875*, 2025.

[18] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[19] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in neural information processing systems*, volume 30, 2017.

[20] Kaiyue Wen, Zhiyuan Li, Jason S. Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape perspective. *ArXiv*, abs/2410.05192, 2024.

[21] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

[22] Seungwoo Kim, Khai Loong Aw, Klemen Kotar, Cristobal Eyzaguirre, Wanhee Lee, Yunong Liu, Jared Watrous, Stefan Stojanov, Juan Carlos Niebles, Jiajun Wu, et al. Taming generative video models for zero-shot optical flow extraction. *arXiv preprint arXiv:2507.09082*, 2025.

[23] Stefan Stojanov, David Wendt, Seungwoo Kim, Rahul Venkatesh, Kevin Feigelis, Jiajun Wu, and Daniel LK Yamins. Self-supervised learning of motion concepts by optimizing counterfactuals. *arXiv preprint arXiv:2503.19953*, 2025.

[24] Rahul Venkatesh, Klemen Kotar, Lilian Naing Chen, Seungwoo Kim, Luca Thomas Wheeler, Jared Watrous, Ashley Xu, Gia Ancone, Wanhee Lee, Honglin Chen, Daniel Bear, Stefan Stojanov, and Daniel Yamins. Discovering and using spelke segments, 2025.

[25] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020.

[26] Yihan Wang, Lahav Lipson Shi, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[27] Zixuan Jiang, Yifan Gao, Junyu Shi, Yujie Wang, Yinlong Zheng, Hujun Bao, and Xiaowei Zhou. Doduo: Learning dense visual correspondence from unsupervised semantic-aware flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[28] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2021.

[29] Daniel M Bear, Kevin Fan, Damian Kuo, Damian Mrowca, Abhishek Gupta, Daniel LK Yamins, and Li Fei-Fei. Unifying (machine) vision via counterfactual world modeling. *arXiv preprint arXiv:2306.01828*, 2023.

[30] Rahul G Venkatesh, Honglin Duan, Kevin Fan, Daniel M Bear, Daniel LK Yamins, and Li Fei-Fei. Counterfactual world modeling for physical dynamics. *arXiv preprint arXiv:2306.01830*, 2023.

[31] Sander Dieleman. Generative modelling in latent space, April 2025. Blog post.

[32] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. *arXiv preprint arXiv:2312.00785*, 2024.

[33] Federico Perazzi, Jordi Pont-Tuset, Benjamin McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] Anand Bhattad, Konpat Preechakul, and Alexei A. Efros. Visual jenga: Discovering object dependencies via counterfactual inpainting, 2025.

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[37] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[38] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[40] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[41] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[42] Irina Higgins, Loic Matthey, Arka Pal, et al. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

[44] Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[45] Felix Stahlberg, Christopher Bryant, and Bill Byrne. Neural grammatical error correction with finite state transducers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4033–4039, Minneapolis, MN, 2019.

[46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, and *et al.* Chain-of-thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022.

[47] Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *CoRR*, abs/2009.03393, 2020.

[48] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *CoRR*, abs/2304.03442, 2023.

[49] Michael Ahn, Anthony Brohan, Noah Brown, Justin Fu, Bob McGrew, Sergey Levine, Karol Hausman, and Chelsea Finn. Do as i can, not as i say: Grounding language in robotic affordances. In *Robotics: Science and Systems (RSS)*, 2022.

[50] Anthony Brohan, Noah Brown, Brian Ichter, Karol Hausman, Alex Irpan, Yevgen Chebotar, Sergey Levine, Vincent Vanhoucke, and Quan Vuong. RT-2: Vision–language–action models transfer web knowledge to robotic control. *CoRR*, abs/2307.15818, 2023.

[51] Danny Driess, Fei Xia, Michael Janner, Sergey Levine, Karol Hausman, and *et al.* PaLM-E: An embodied multimodal language model. *CoRR*, abs/2303.03378, 2023.

[52] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *Conference on Robot Learning (CoRL)*, pages 37–46, 2018.

[53] Ashvin Nair, Vitchyr Pong, Michaël Chang, and Sergey Levine. R3M: A universal visual representation for robot manipulation. *CoRR*, abs/2203.12601, 2022.

[54] Christopher Krupenye, Fumihiro Kano, Satoshi Hirata, Josep Call, and Michael Tomasello. Great apes anticipate that other individuals will act according to false beliefs. *Science*, 354(6308):110–114, 2016.

[55] Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental Science*, 10(1):89–96, 2007.

[56] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[57] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. Generative adversarial nets. In *NeurIPS*, 2014.

[58] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[59] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[60] Mark Chen, Alec Radford, Rewon Child, et al. Generative pretraining from pixels. In *ICML*, 2020.

[61] Arnaud Pannatier Ning, Soufiane Hayou, Aishwarya Ghosh, Stefanie Jegelka, Johannes Kirschner, and Nikolay Malkin. Sigma-gpts: A new approach to autoregressive models. *arXiv preprint arXiv:2404.09562*, 2024.

[62] Zichen Gao, Tianhong Wang, Lijun Fan, Zangwei Liu, Yujie Tang, Junmin Chen, Jun Zhu, and Bo Zhang. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

[63] Xiaoran Han, Ye Zhang, Yizhe Chen, et al. Xl-vae: Scaling up variational autoencoders for image generation with extra layers. *arXiv preprint arXiv:2024.xxxxx*, 2024.

[64] Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proc. 7th Int. Conf. on Machine Learning*, pages 216–224, 1990.

[65] Marc Peter Deisenroth and Carl Edward Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proc. 28th Int. Conf. on Machine Learning (ICML)*, pages 465–472, 2011.

[66] Carlos E. Garcia, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[67] David Ha and Jürgen Schmidhuber. World models. *arXiv:1803.10122*, 2018.

[68] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Learning latent dynamics for planning from pixels. In *ICML*, 2019.

[69] Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *CoRR*, abs/2206.04114, 2022.

[70] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model-based reinforcement learning for atari. *CoRR*, abs/1903.00374, 2019.

[71] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[72] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Proc. 6th Conf. on Robot Learning (CoRL)*, pages 2226–2240, 2023.

[73] Klemen Kotar, Aaron Walsman, and Roozbeh Mottaghi. ENTL: Embodied navigation trajectory learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10863–10872, October 2023.

[74] Brendan J Frey and Nebojsa Jojic. Learning graphical models of images, videos and their spatial transformations. *arXiv preprint arXiv:1301.3854*, 2013.

[75] Matthew J Beal, Nebojsa Jojic, and Hagai Attias. A graphical model for audiovisual object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):828–836, 2003.

[76] Manavender R Malgireddy, Ifeoma Inwogu, and Venu Govindaraju. A temporal bayesian model for classifying, detecting and localizing activities in video sequences. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 43–48. IEEE, 2012.

[77] McCoy R Becker, Alexander K Lew, Xiaoyan Wang, Matin Ghavami, Mathieu Huot, Martin C Rinard, and Vikash K Mansinghka. Probabilistic programming with programmable variational inference. *Proceedings of the ACM on Programming Languages*, 8(PLDI):2123–2147, 2024.

[78] Nishad Gothoskar, Marco Cusumano-Towner, Ben Zinberg, Matin Ghavamizadeh, Falk Pollok, Austin Garrett, Josh Tenenbaum, Dan Gutfreund, and Vikash Mansinghka. 3dp3: 3d scene perception via probabilistic programming. *Advances in Neural Information Processing Systems*, 34:9600–9612, 2021.

[79] Marco F Cusumano-Towner, Feras A Saad, Alexander K Lew, and Vikash K Mansinghka. Gen: a general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th acm sigplan conference on programming language design and implementation*, pages 221–236, 2019.

[80] Lionel Wong, Gabriel Grand, Alexander K Lew, Noah D Goodman, Vikash K Mansinghka, Jacob Andreas, and Joshua B Tenenbaum. From word models to world models: Translating from natural language to the probabilistic language of thought. *arXiv preprint arXiv:2306.12672*, 2023.

[81] Lionel Wong, Katherine M Collins, Lance Ying, Cedegao E Zhang, Adrian Weller, Tobias Gersternberg, Timothy O'Donnell, Alexander K Lew, Jacob D Andreas, Joshua B Tenenbaum, et al. Modeling open-world cognition as on-demand synthesis of probabilistic models. *arXiv preprint arXiv:2507.12547*, 2025.

[82] Tim Maudlin. *The metaphysics within physics*. Oxford University Press, 2007.

[83] Nancy Cartwright. *Causal Powers: What are They?, why Do We Need Them? what Can be Done with Them and what Cannot?* Contingency and dissent in science project. Centre for the philosophy of the . . . , 2007.

[84] Tobias Gerstenberg. Counterfactual simulation in causal cognition. *Trends in Cognitive Sciences*, 28(10):924–936, 2024.

[85] Judea Pearl. 3. the foundations of causal inference. *Sociological Methodology*, 40(1):75–149, 2010.

[86] Richard S. Sutton. The bitter lesson. Blog post, 2019.

[87] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[88] Feras A Saad, Marco F Cusumano-Towner, Ulrich Schaechtle, Martin C Rinard, and Vikash K Mansinghka. Bayesian synthesis of probabilistic programs for automatic data modeling. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–32, 2019.

[89] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

[90] Michael Y Li, Emily B Fox, and Noah D Goodman. Automated statistical model discovery with language models. *arXiv preprint arXiv:2402.17879*, 2024.

[91] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

[92] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

[93] Rosa Cao and Daniel Yamins. Explanatory models in neuroscience, part 2: Functional intelligibility and the contravariance principle. *Cognitive Systems Research*, 85:101200, 2024.