ROBOJEPA: SCALING ROBOTIC LATENT WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models have shown a remarkable ability to predict future states and to plan in the real world. In practice, however, we lack a principled way to estimate how their capabilities scale with model size, data, and compute, an open problem that slows progress in the field. In this work we present RoboJEPA, a world model based on the Joint Embedding Predictive Architecture (JEPA) and trained on a large-scale dataset spanning 12 robotic embodiments. We show that RoboJEPA's imagination error, the error of its latent rollouts, follows a second-order power law in compute, allowing us to predict model quality well beyond the scale at which the law is fit. We further show that downstream robotic planning performance improves predictably with compute, and that imagination error is strongly correlated with it, making it a reliable proxy for real-robot evaluation. Finally, we demonstrate that latent world models can be deployed zero-shot as robotic agents, planning toward a single goal image to solve tasks requiring long-horizon planning on real hardware. We will release all model checkpoints together with our training and robot deployment code. To our knowledge, this is the first work to establish scaling laws for multi-embodiment robotic world models trained on real robot data, and RoboJEPA, at 8B parameters, is the largest JEPA predictor model trained to date.

1 INTRODUCTION

Scaling has been one of the primary engines of recent progress in artificial intelligence. What makes scaling particularly actionable is *predictability*: empirical scaling laws relate a model's loss to the amount of data, parameters, and compute used to train it, so that the return on an additional order of magnitude of compute can be forecast before it is spent (Kaplan et al., 2020; Hoffmann et al., 2022). This predictability turns scaling from a gamble into a science, it tells practitioners how large a model to train for a given budget, how much data to pair it with, and when returns will diminish. Scaling laws of this kind have been established across language models (Kaplan et al., 2020; Hoffmann et al., 2022), generative modeling of images, video, and other modalities (Henighan et al., 2020), or semantic classification of images (Zhai et al., 2022). Comparatively little is known about whether, and how, the same scaling predictability holds for models of the physical world.

We study this question for *world models* in robotics. A world model is a learned model of environment dynamics that an agent can roll out to imagine future states and plan (Ha & Schmidhuber, 2018; Hafner et al., 2023). We focus on latent world models that predict and plan in a latent representation space (Hafner et al., 2020; LeCun, 2022; Assran et al., 2025). Latent world models have shown the ability to plan toward goals specified at test time without the need of a task-specific policy (Assran et al., 2025). Previous latent world models are typically trained at a single scale, and how their quality changes with data, parameters, and compute has not been characterized (Assran et al., 2025; Hafner et al., 2025; Zhang et al., 2026). Other studies, focusing on scaling, either target generative video and interactive-environment models outside of robot control (Bruce et al., 2024; Google DeepMind, 2024; Hu et al., 2023), or scale *policies* rather than world models (Lin et al., 2025; Sartor & Thompson, 2024; Ai et al., 2025). In contrast, our goal is to establish scaling laws for latent world models. This matters particularly in robotics, where high-quality interaction data is scarce and computationally expensive to collect: knowing which capabilities a world model will have at a given training-compute budget lets practitioners decide whether to invest in a larger model, more demonstrations, or more compute, and reveals when a data budget has been saturated.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To this end, this work introduces **RoboJEPA**, a family of action-conditioned latent joint-embedding predictive world models. We keep the training recipe deliberatly simple to focus on the scaling behavior. **RoboJEPA** is a feed-forward transformer that predicts the next latent state deterministically in a single forward pass. It is trained with a plain ℓ_1 regression loss in the frozen encoder's latent space. **RoboJEPA** operates in the representation space of a frozen V-JEPA 2.1 encoder (Assran et al., 2025; Mur-Labadia et al., 2026) and is trained on a unified mixture of 23 public manipulation datasets spanning 12 embodiments and 26 different action spaces. We study model scaling of the *world model*, ranging parameter count from 22M up to 8B, while holding the representation encoder fixed, so our laws characterize how a plannable dynamics model scales on top of a shared representation. We build scaling laws estimating the compute-efficient frontier of the forward-prediction evaluation loss as a function of the training compute (Kaplan et al., 2020; Pearce & Song, 2024).

Our central finding is that robotic world-model quality scales predictably with compute. First, the latent forward-prediction loss on out-of-distribution data follows a scaling law $L(C) = A\,C^{\alpha-\gamma\ln C} + E$, where C is the training compute, E is the irreducible loss, and A, α, γ are scaling-law parameters that we estimate alongside E. This holds on both real-robot holdout (DROID (Khazatsky et al., 2024)) data and simulated holdout (RoboCasa (Nasiriany et al., 2026)) data. Second, this improvement on the offline data transfers to downstream planning performance on a robot: on a suite of RoboCasa planning tasks, downstream success improves predictably with training compute, so that a world model's latent forward-prediction loss is a strong predictor of its online planning performance. For example, only models trained with more than 10^{22} FLOPs of training compute (at sizes 2B, 4B, and 8B) can solve object-interaction tasks in RoboCasa via non-greedy planning, while smaller models trained with less compute cannot (see Figure 3d).

We deploy RoboJEPA on real and simulated robots, planning toward a single goal image that defines the final state of the environment. Across two robot platforms, we perform planning in over $50{,}000$ evaluation episodes. We keep the planning hyperparameters fixed for each robot, varying only the planning horizon for harder tasks. We perform planning on a set of real world manipulation tasks that include grasping, lifting, and pick-and-place. In simulation, our planning spans tasks of increasing difficulty from reaching to object pushing. We observe an improvement trend in the success rate on the real robot as we increase the model size. Qualitatively, we also observe that the behavior of the planning policies changes as the model grows: for example, in object-manipulation tasks, only models with 2B parameters or keep the gripper closed after grasping to complete the task, while smaller models often accidentally open it, dropping the object.

Finally, to support open research and reproducibility, we will release all model checkpoints together with the training and robot-deployment code. To the best of our knowledge, this is the first work to establish compute-optimal scaling laws for action-conditioned robotic world models trained on large-scale, multi-embodiment real robot data, and the first scaling law for JEPA with verified extrapolation. We also train the largest JEPA *predictor* to date, at 8B parameters.

2 Method and Data

To study the scaling behavior of robotic world models, we require a model family that scales with training compute. To this end we introduce RoboJEPA, a family of action-conditioned latent world models . The goal of RoboJEPA models is to predict how the latent features of a scene will evolve under the robot's actions. Formally, at timestep t, given a history of past visual features $z_{\le t}$ (where z_t is the latent encoding of a given visual input x_t), proprioceptive states $s_{\le t}$, and actions $a_{\le t}$, our models are functions g_θ that deterministically predict the features of the next frame in one neural network forward pass:

$$\hat{z}_{t+1} \;=\; g_\theta\big([z_{\le t}],\, a_{\le t},\, s_{\le t}\big). \tag{1}$$

Actions are either state deltas, i.e. $s_{t+1} = s_t + a_t$, or absolute robot commands, in which case $s_t = 0$. Feeding predictions back as inputs lets us roll the model out autoregressively over future actions, e.g.

$$\hat{z}_{t+2} \;=\; g_\theta\big([z_{\le t}, \hat{z}_{t+1}],\, a_{\le t+1},\, s_{\le t+1}\big), \quad \hat{z}_{t+3} \;=\; g_\theta\big([z_{\le t}, \hat{z}_{t+1}, \hat{z}_{t+2}],\, a_{\le t+2},\, s_{\le t+2}\big), \quad \dots \tag{2}$$

This allows us to autoregressively roll out the world model to predict the effect of a sequence of actions on the latent features of the scene, and consequently to *plan*.

Planning in RoboJEPA is specified by a goal image encoded once into goal features z_g, and at every timestep t we search for the H-step action sequence whose imagined latent rollout comes closest to

the goal:

$$a^{\star}_{t:t+H-1} \;=\; \arg\min_{a_{t:t+H-1}} \left\| \hat{z}_{t+H}(a_{t:t+H-1}) - z_g \right\|_1, \tag{3}$$

where $\hat{z}_{t+H}$ is obtained by rolling g_θ forward H steps from the current context under the action sequence $a_{t:t+H-1}$. We optimize Equation (3) with the cross-entropy method (a concrete instance is given in Section 2.2 and appendix G).

The rest of this section describes the design and training of this family of world models and how to use them for planning, with the goal of subsequently studying their scaling behavior.

2.1 RoboJEPA Architecture

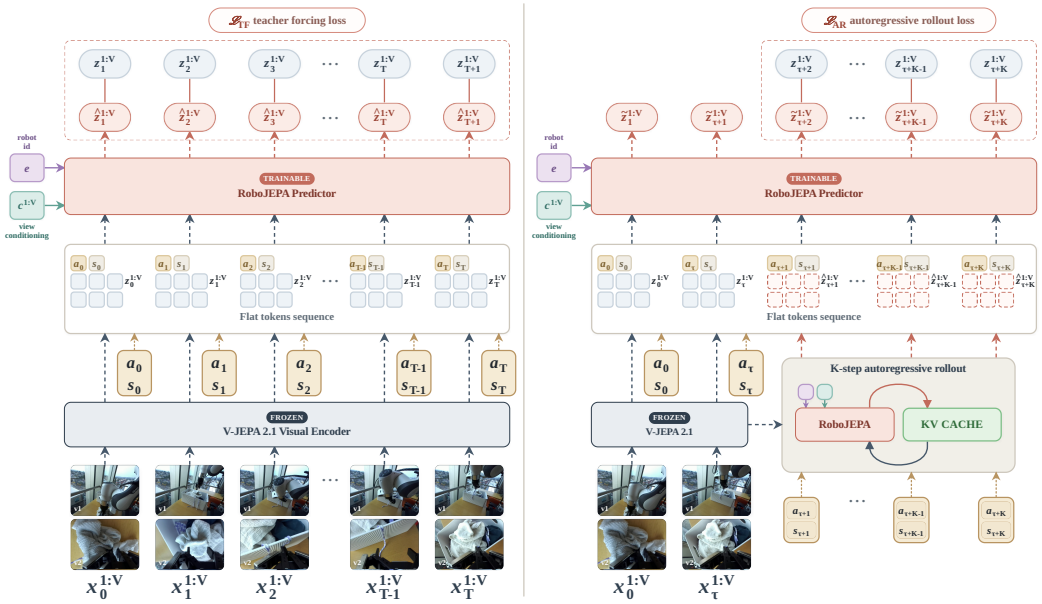


Figure 1: RoboJEPA training. The frozen V-JEPA 2.1 encoder f (Mur-Labadia et al., 2026) maps the input multiview frames $x_t^{(1:V)}$ to a grid of features. The predictor g_θ consumes per-step sequences of multiview visual ($z_t^{(1:V)}$), action (a_t) and proprioceptive state (s_t) tokens, conditioned on view and embodiment identifiers, and predicts the next-step features $\hat{z}_{t+1}$. Training minimises two losses: **(left) the teacher-forcing loss**, the per-token ℓ_1 distance between predicted features and the next-step features from the encoder (Equation (5)); and **(right) the rollout loss**, which attempts to predict the same target sequence as the teacher-forcing loss, but given fewer input frames. The model is given the full training sequence of actions but only a prefix of the encoded features, and predicts the remaining features autoregressively, with a KV-cache. This way the model learns to make accurate long term predictions.

The family of RoboJEPA world models builds on the action-conditioned recipe of V-JEPA 2 (Assran et al., 2025) and V-JEPA 2.1 (Mur-Labadia et al., 2026). This subsection introduces the architecture of the RoboJEPA predictor. The full list of differences between RoboJEPA and V-JEPA 2-AC is summarised in Table 4 and detailed in Appendix B.

Scaling world models across robot datasets is challenging because each dataset comes with a different number of camera views, proprioceptive state size, and controller type and frequency. Therefore, we aim to build a world model architecture that can work with any combination of views, controller type, and proprioceptive information. At each timestep t, a frozen V-JEPA 2.1-G encoder maps up to V simultaneous camera views at resolution $H \times W$, each spanning T frames, denoted $x_t^{(1:V)} = (x_t^{(1)}, \ldots, x_t^{(V)})$, to the latent features $z_t^{(1:V)} = (z_t^{(1)}, \ldots, z_t^{(V)})$, where each $z_t^{(v)}$ is an $H_p \times W_p$ grid of D_e-dimensional patch embeddings with $D_e = 1664$ and $H_p = H/16$, $W_p = W/16$. We use V-JEPA 2.1 as the encoder because early in the work we found that it leads to

Table 1: Compressed view of the 23-dataset, 26-action-space mixture. Video hours sum the durations of all recorded videos. Action hours measure the total duration of teleoperation or policy execution, counted once regardless of the number of cameras. The per-dataset breakdown with resolutions and source FPS is in Table 7; sampling weights are in Table 8; and semantic-camera mappings are in Table 9. We report dataset sizes that we were able to collect and verify, which may be smaller than those reported in the original papers.

Dataset group	Episodes	Video hours	Action hours	Embodiments
DROID (Khazatsky et al., 2024)	92k	414	138	Franka
Roboset (Bharadhwaj et al., 2024)	73k	1244	380	Franka (Roboset)
RoboMind (Wu et al., 2024b)	35.3k	276	98	Franka, AgileX
LeRobot (Cadene et al., 2024)	21.5k	182	93	SO-101
RoboCasa365 (Nasiriany et al., 2026)	32k	482	482	Franka + Omron (sim)
1X (1X Technologies, 2024)	23.5k	90	90	1X humanoid
AgiBot World (AgiBot-World-Contributors et al., 2025)	167.5k	8036	2734	AgiBot dual-arm, humanoid
Open-X Embodiment (Open X-Embodiment Collaboration et al., 2023)[1]	2.4265M	4298	2677	8 platforms
Total	**2.8713M**	**15,022**	**6,692**	12 platforms / 26 ids

successful and predictable scaling of the world model, whereas V-JEPA 2 does not (we hypothesize why in Appendix N). Alongside these features, each timestep provides the action $a_t \in \mathbb{R}^{d_a}$ and proprioceptive state $s_t \in \mathbb{R}^{d_s}$, while each input is associated with a robot type operating in a particular action space, identified by the *robot id* $e \in \mathcal{E}$, and a per-view camera type $c^{(v)} \in \mathcal{C}$. Here, $\mathcal{E}$ and $\mathcal{C}$ are finite sets of known robot ids and camera types. Next we describe how we tokenize inputs, and the world model architecture.

Tokenization. We encode each modality separately and interleave the results into a single flat sequence. We refer to these encoded multimodal vectors as tokens. The motivation of our tokenization scheme is to let the model learn all temporal dependencies only from a structured sequence of tokens. To tokenize each $z_t^{(v)}$, we project each patch from D_e to the world-model width D with a single linear layer, giving a visual token tensor of shape $[T, V, H_p, W_p, D]$. Actions and proprioceptive states are mapped to D by linear encoders (specific to robot id) applied to a_t and s_t.

We interleave these tokens by timestep,

$$\left(z_1^{(1:V)},\, a_1,\, s_1\right),\ \left(z_2^{(1:V)},\, a_2,\, s_2\right),\ \ldots,\ \left(z_T^{(1:V)},\, a_T,\, s_T\right), \tag{4}$$

so that each timestep contributes VH_pW_p visual tokens followed by one action and one state token. Beyond this ordering we impose no structure: the model consumes a flat sequence and recovers spatio-temporal and view structure from the position encodings and external conditioning information, such as the robot id and view type.

World Model Architecture. RoboJEPA performs the next embedding prediction via a deterministic model. RoboJEPA uses the vision transformer (Dosovitskiy et al., 2021) as the backbone and we parameterize the prediction such that each spatial input token of $z_t^{(1:V)}$ predicts the corresponding spatial token of $z_{t+1}^{(1:V)}$. We use the frame-level causal attention mask so that queries that correspond to timestep t have only access to keys and values at step t or earlier. The model stacks L blocks that form a standard residual stream of ViT. The architecture of each block follows the standard recipe for modern variants of ViTs: we use pre-normalization, Rotary Position Encoding (Su et al., 2024), RMSNorm normalization (Zhang & Sennrich, 2019), GELU feed-forward block (Hendrycks & Gimpel, 2016), and QK-normalization (Dehghani et al., 2023). We found QK-normalization to be of particular importance for training stability when scaling the dataset size and the model size. We use 4D RoPE factorizing positions over views, time, height, and width. The view conditioning is a per-layer learnable bias added across the full head width of visual queries and keys after RoPE. We use an attention mask that is causal across timesteps and local to the past 8 timesteps. See Appendix B for more details.

[1]We use 9 datasets of Open-X Embodiment (Open X-Embodiment Collaboration et al., 2023): RT-1 (Brohan et al., 2023), Bridge (Ebert et al., 2022), Bridge V2 (Walke et al., 2023), QT-Opt (Kalashnikov et al., 2018), MT-Opt (Kalashnikov et al., 2021), RoboNet (Dasari et al., 2019), Language-Table (Lynch et al., 2023), BC-Z (Jang et al., 2022), FMB (Luo et al., 2024).

Outputs. After feeding the entire sequence of tokens through the backbone, we project tokens that correspond to $z_{1:T}^{(1:V)}$ back to the encoder space. We map the output from D dimensions to D_e dimensions and apply layer normalisation. The idea is that since both input and output features are normalized, this strenghtens the learning signal and lowers the gap between the training and the autoregressive generation. We do not use any learnable modules to predict states and we simply compute the proprioceptive state s_t either through integration, i.e. $s_{t+1} = s_t + a_t$, or $s_t = 0$ (for example when working with commanded actions with non-trivial robot dynamics). For qualitative visualization, we train a diffusion decoder to convert predicted latent features into videos (see Appendix E for decoder details and Appendix F for visualization examples).

Training objective. We train the world model to predict the next embedding and, to minimize exposure bias, we additionally train it on its own outputs over potentially long horizons. Our objective therefore has two loss components. The first component trains the model to predict every next embedding along a sequence of encoded multiview frames. The second component uses the same objective, but the model is given only a prefix of fewer encoded frames (but the entire sequence of actions); it then predicts the missing embeddings autoregressively (Equations (1) and (2)) and is trained on them with the same objective. We use a key-value cache to speed up this autoregressive prediction (Section 2.2 and appendix I). While the first loss component ensures that the model predicts the representations of a fixed encoder, the second makes the world model robust to its own errors and suppresses error accumulation. Both components use the same per-token ℓ_1 objective, and the total loss is their average:

$$\mathcal{L}_{\mathrm{TF}} = \sum_{t,v} \left\| \hat{\boldsymbol{z}}_{t+1}^{(v)} - z_{t+1}^{(v)} \right\|_1, \quad \mathcal{L}_{\mathrm{AR}} = \sum_{t,v} \left\| \tilde{\boldsymbol{z}}_{t+1}^{(v)} - z_{t+1}^{(v)} \right\|_1, \quad \mathcal{L} = \tfrac{1}{2}\mathcal{L}_{\mathrm{TF}} + \tfrac{1}{2}\mathcal{L}_{\mathrm{AR}}, \tag{5}$$

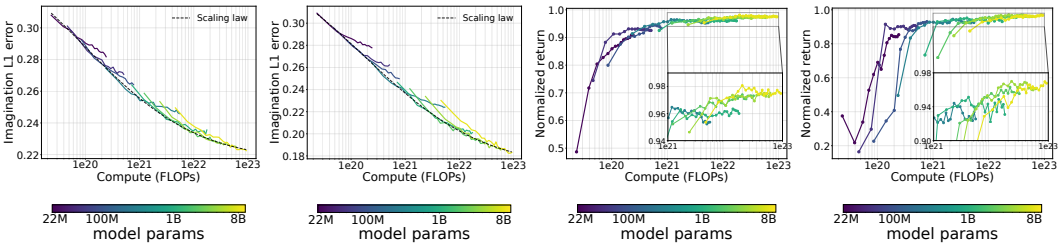
where $\hat{\boldsymbol{z}}_{t+1}^{(v)}$ is the teacher-forced one-step prediction and $\tilde{\boldsymbol{z}}_{t+1}^{(v)}$ is the autoregressive prediction rolled out from a shorter prefix, both compared against the encoder feature $z_{t+1}^{(v)}$. The full form of the loss (with the per-frame mask and patch normalization), the rollout schedule, and the optimizer are described in Section 2.2 and appendix D.

2.2 Data, Training and Inference

Data. Training a world model that generalizes across robot embodiments and scenes requires both scale and diversity. We assemble 23 public manipulation datasets spanning 12 robot platforms—single- and dual-arm robots and humanoids—across real and simulated data. The compiled dataset contains 15,022 hours of video, including 6,692 hours synchronized with actions; some demonstrations are recorded from multiple cameras. Some platforms record with only one camera (e.g. most of Open-X Embodiment data), whereas others use two or three cameras (e.g. DROID and RoboSet). In addition, the datasets were collected with different robot action spaces, camera poses, sensor resolutions, and frame rates. Therefore, to build a data-scalable world model, we unify our training datasets and formulate the architecture and training recipe to be general enough to work with any robotic dataset. Table 1 summarizes the mixture, and Appendix C provides further details about its action spaces, unification, and sampling.

Training. We train a series of RoboJEPA models that range in size from 22M to 8B parameters and in training compute from 2×10^{19} to 9.5×10^{22} FLOPs. For all models, we use AdamW (Loshchilov & Hutter, 2019) with a weight decay of 0.1 and global gradient-norm clipping at 1.0. We linearly warm up the learning rate, hold it constant at a peak of 5×10^{-5} for most of training, and then linearly decay it to zero over a short cooldown phase at the end (Hu et al., 2024). This allows us to overlap the flat learning-rate stage across models trained with different amounts of compute: we run the flat stage only once and perform learning-rate annealing from intermediate checkpoints. For the detailed description of the training configuration and hyperparameters, please refer to Appendix D.

Inference and Planning. RoboJEPA performs planning by predicting the future in the latent space and searching over the action sequence that brings that future to the desired goal. We employ receding-horizon CEM (Rubinstein, 1997; de Boer et al., 2005) as our planning algorithm with image goals; importantly, we use only a single image goal that represents the final state of the task, and we do not use subgoals. For a more efficient planning, we use compiled KV-cache inference (see Appendices I and J for details) and parallelize CEM across several GPUs. This way, we make the planning fast enough to be practical: while it is not real-time, it takes only several seconds to perform a multi-step planning even for an 8B model.



(a) DROID. (b) RoboCasa. (c) Pose reach. (d) Object reach.

Figure 2: **World-model scaling laws and downstream planning.** (a,b) Out-of-distribution forward-prediction L_1 error versus training compute on the DROID holdout and RoboCasa target scenes. The compute-optimal frontier follows the second-order power law $L(C)=E+A\,C^{\alpha-\gamma\ln C}$ (dashed). (c,d) Normalized return for pose reaching and pose reaching while holding an object in unseen RoboCasa kitchens; insets magnify the saturation region. Color indicates model size, from 22M to 8B parameters. Full-resolution versions of panels (a)–(d) appear in Figures 30 to 33, respectively.

3 World Model Scaling Laws

Our goal is to empirically find the concrete functional form of the scaling law that allows us to extrapolate predictions about the performance of the RoboJEPA model trained at larger scales. We consider four candidate functional forms. First, the standard power law $L(C) = E + A\,C^{\alpha}$ (Henighan et al., 2020), whose exponent α is constant. Second, a generalization whose exponent itself changes linearly with compute, $L(C) = E + A\,C^{\alpha-\gamma\ln C}$; we refer to the former as a *first-order* and to the latter as a *second-order* power law[1]. Finally, the Broken (Caballero et al., 2022) and Unified (Caballero et al., 2026) Neural Scaling Laws. Here, C is the total training compute (in FLOPs) and $L(C)$ is the world model's forward-prediction error, i.e. the per-token ℓ_1 distance between the predicted and the encoder features, averaged over a 9-step latent rollout on held-out data. Among these four forms, we find that RoboJEPA predictably follows the second-order power law.

Setup. We evaluate each model trained at each compute budget by its forward-prediction L_1 error, $L(C)$, on held-out data. Importantly, this is the same per-token L_1 objective the model is trained on and that the planner minimizes at deployment (Equation (5)). We report $L(C)$ on two embodiments, a DROID holdout dataset collected on a real robot and a simulated RoboCasa suite. Evaluation is always on scenes unseen during training, always uses two views: one side camera (left or right) and one wrist camera, and predicts nine future steps from an initial observation under the recorded action sequence, giving a ten-step sequence in total. Further evaluation details are provided in Appendix K. For each model at each training duration we compute the total training compute C exactly, by counting the FLOPs of every operation in one training step and multiplying by the number of steps taken across all stages. We plot the prediction error for every model and training duration against its total training compute. At each compute budget, we retain the lowest error across model sizes. These best-achievable errors form the compute-optimal frontier as a function of the training compute. Finally, we fit each candidate curve to the frontier points of the smaller models (22M–2B) and select as our scaling law the one that best extrapolates to the held-out 4B and 8B models.

Results. Table 2 compares the four candidate curves by how well a fit on the smaller models (22M–2B) extrapolates to the held-out 4B and 8B models. The second-order power law extrapolates best, with a held-out error of 0.6 and 1.4×10^{-3} on DROID and RoboCasa respectively, roughly 2–3$\times$ lower than the standard power law (2.0 and 3.7×10^{-3}) and below both BNSL and UNSL. We therefore adopt it as our scaling law. Figure 2 plots the forward-prediction error against training compute for all models. The lower envelope across model sizes forms the compute-optimal frontier, and no model size lies on the frontier entirely. In addition, Figure 2 plots the scaling law computed on all models as the dashed line. Interestingly, the two best candidates, the second-order law and BNSL, estimate nearly the same irreducible error, ≈ 0.2 on DROID and ≈ 0.17 on RoboCasa, indicating that RoboJEPA is close to saturation on both evaluations under its current training and data budget.

[1] It is easy to show that, for the latter, the dependency between $\ln(L-E)$ and $\ln C$ is a quadratic function of $\ln C$.

Table 2: Variants of parametric curves as scaling-law candidates for the L_1 rollout error of RoboJEPA world models. Curve fits use models sized 22M–2B parameters; we report held-out extrapolation error at 4B and 8B for DROID and RoboCasa. The BNSL and UNSL functional forms and their fitted parameter values are in Appendix K.

Parametric Curve	Curve parameters		Extrapolation Error (4B and 8B), $\times 10^{-3}$	
	DROID	RoboCasa	DROID	RoboCasa
$L(C) = E + A\,C^{\alpha}$	E=0.182, A=108.8, α=−0.151	E=0.075, A=23.2, α=−0.103	2.0	3.7
$L(C) = E + A\,C^{\alpha-\gamma \ln C}$	E=0.218, A=7.96×10^{-19}, α=1.91, γ=0.02308	E=0.1709, A=1.66×10^{-17}, α=1.761, γ=0.02105	**0.6**	**1.4**
BNSL (Caballero et al., 2022), n=1	Appendix K.1		1.0	1.8
UNSL (Caballero et al., 2026), n=1	Appendix K.2		1.4	5.7

3.1 Downstream Planning Scaling

Having identified an improvement trend in offline performance of world models, we hypothesize that this improvement translates to the world model's physical understanding of the new environment in which it operates. For example, is the action conditioning accurately mapped to the correct movements of the robot in the representation space? Are the grasping and object interaction actions correctly represented by the world model in response to incoming action? We test this hypothesis via online planning with image goals. Section 3 introduced the predictable improvement in offline forward-prediction, and we verify that this improvement translates into a better success rate.

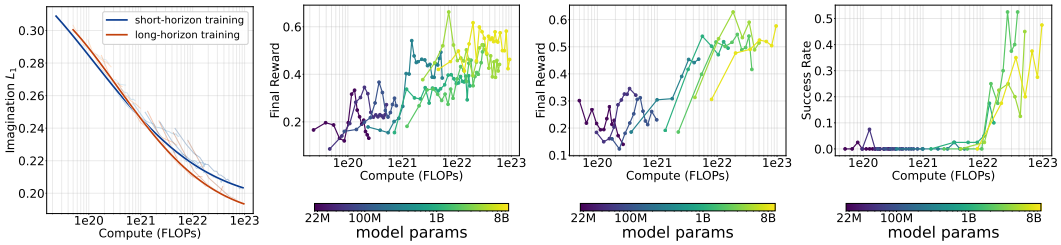
To test this hypothesis, we build a set of interactive environments on top of RoboCasa 365 (Nasiriany et al., 2026), each isolating one basic capability, and test whether more training compute improves it. In this section we focus on *greedy* tasks, which can be solved by one-step planning where simply moving toward the goal at every step is enough. (i) *3D understanding* (pose reaching): reach a target end-effector pose in as few steps as possible. (ii) *Object understanding* (pose reaching while holding an object): reach a target pose while holding a grasped object, so the planner must additionally search over gripper actions to avoid dropping it. Harder tasks that require non-greedy, longer-horizon planning — such as moving around an obstacle or pushing an object to a target — are studied in Section 3.2. Full environment and reward details are in Appendix H.4.

We use the same range of RoboJEPA checkpoints as in Section 3 and run each of them in our RoboCasa environments with closed-loop planning (Mayne et al., 2000), where the model re-plans toward the goal image after every executed action (full protocol in Appendix H.4). Figure 2 shows the evaluation results, measured by the simulator's normalized cumulative reward on the performed trajectory (Appendix H.4). Just like with the offline performance, we see a clear improvement trend of the downstream performance as we scale the training compute. For both tasks, models form a clear performance frontier, with larger models surpassing smaller ones as the training compute grows. For example, the simplest end-effector 3D movement capabilities emerge after around 10^{20} FLOPs while the basic capability of moving while holding an object emerges at around 3×10^{20} FLOPs of training, both of which correspond to training 50M–100M-parameter models on the full multi-embodiment dataset. Due to the simplicity of these tasks, performance saturates early. Therefore, for building robotic world models, we recommend first building downstream evaluations that reflect the desired capabilities, then estimating the minimal training compute required to achieve them.

3.2 Better Scaling for Long Horizon Planning

With the established transfer from offline performance to online greedy performance in Sections 3 and 3.1, we now ask whether the same offline-to-online connection holds for *non-greedy*, longer-horizon tasks that cannot be solved in one step. We design two such tasks in RoboCasa. The first is an *obstacle-reach* task: the goal is again to reach a target end-effector position, but we place a large object (a kitchen appliance) as an obstacle so that a direct straight-line motion cannot reach the goal. The second is an *object-push* task, where an object is placed on the table in front of the robot and the goal is to move it toward a target position, which requires the arm to descend to the object, push it, and ascend back. Full task definitions are in Appendix H.4.

We first deploy the models trained with the recipe of Section 3 on the obstacle-reach task. Figure 3b shows that the final reward correlates with training compute, but the correlation is weak and no clean compute frontier emerges: smaller models often beat larger ones (for example, the 300M planner reaches a higher final reward than the 1B one). Unlike the greedy tasks of Section 3.1, predictable scaling does not emerge for longer-horizon tasks, which we attribute to a mismatch between training and deployment. Planning on these tasks relies on accurate *long-horizon* prediction, yet the models were trained with only a short autoregressive rollout ($K = 2$ steps). We therefore align training with how the model is used at test time and increase the training rollout to $K = 10$ steps during the cooldown, together with higher-resolution inputs. We refer to the two cooldown recipes as *short-horizon prediction training* ($K = 2$, 240×320) and *long-horizon prediction training* ($K = 10$, a mix of 240×320 and 480×640).



(a) Error scaling. (b) Short rollout. (c) Long rollout. (d) Push success.

Figure 3: **Long-horizon prediction training improves scaling and downstream planning.** (a) Imagination error versus compute for the short- and long-horizon cooldown recipes. (b,c) Obstacle-reach final reward for short-horizon ($K = 2$) and long-horizon ($K = 10$) models. (d) Object-push success rate for the long-horizon models. Full-resolution versions of panels (a)–(d) appear in Figures 34 to 37, respectively.

Figure 3a compares the two recipes by their scaling laws on DROID and RoboCasa. Long-horizon prediction training is more computationally expensive per step, but it becomes more compute-efficient beyond about 10^{21} FLOPs of total training compute (around 300M–1B parameters at the compute-optimal frontier) and reaches a substantially lower forward-prediction error. It also lowers the *irreducible* error of the fit, indicating a genuine reduction in error accumulation.

Deploying the non-greedy planning models on the obstacle-reach task (Figure 3c) significantly improves the trend: the final reward now strongly follows the predictable scaling pattern. Small models (below 300M parameters) cap at around 0.3, which corresponds to the end-effector remaining stuck on the wrong side of the obstacle. The obstacle-avoidance capability then takes off around 10^{21} FLOPs and saturates around 10^{22} FLOPs. This is much later than the end-effector capabilities of Section 3.1, which emerged around 10^{20}–3×10^{20} FLOPs, indicating that modeling interaction with the static scene geometry requires considerably more training.

Finally, the object-push task assesses the object interaction understanding (through non-greedy planning) by the world model: because the robot's start and goal poses are identical (so the side and wrist views change only through the object's position), the model must understand that pushing the object is what changes the scene. Figure 3d shows the success rate, defined simply as whether the object moved toward its target position (Appendix H.4). Interestingly, no model achieves a positive success rate until around 10^{22} FLOPs, after which all models begin to succeed, suggesting that fine-grained understanding of surrounding objects emerges after certain training computation regardless of the model size. In summary, these results align in a natural order in which capabilities appear with more compute: first the 3D control of the end-effector, then manipulating a grasped object, then the geometry of the static scene, and finally the dynamics of objects in the scene.

3.3 Real World Robotic Capabilities

Our ultimate goal is to showcase the capabilities of RoboJEPA in real-world settings. As a step towards this goal, we deploy RoboJEPA models ranging from 22M to 8B parameters on the DROID (Khazatsky et al., 2024) platform with a single-arm Franka robot and two cameras: an external camera mounted to the left of the arm and a wrist camera using the standard DROID mount. To deploy RoboJEPA via

planning, we record an image goal on the robot by executing a predefined robot motion and manually placing the object in the grasping position so the robot can interact with it. We provide only a single final image goal, requiring longer-horizon, non-greedy planning. We evaluate three tasks: Grasp, Object Lift, and Pick and Place. We use the same planning hyperparameters across tasks and model sizes, evaluating the highest-compute checkpoint at each size. Setup, task definitions, episode counts, scoring criteria, and planning hyperparameters are detailed in Appendices H.1 to H.3.

To contextualize the RoboJEPA results, we also evaluate the Vision-Language-Action (VLA) models π_0-FAST (Pertsch et al., 2025) and $\pi_{0.5}$ (Black et al., 2025). However, their deployment differs in goal specification (text instructions versus a single goal image) and training (DROID fine-tuning for the VLA baselines versus pretraining on a mixture that includes DROID for RoboJEPA). Neither RoboJEPA nor the VLA baselines receive task-specific fine-tuning. These differences make the VLA results contextual references rather than directly comparable baselines. Figure 4 shows the scaling trend. For each task, all RoboJEPA models, including very small models, demonstrate non-zero success rates. Overall, success rates show a clear trend of improving with model size and generally decreasing with task complexity. This picture does not hold for the reference models, which work best on Pick and Place, the hardest task, but degrade significantly on the others. For example, $\pi_{0.5}$ achieves a 53% success rate on Pick and Place but 0% on Object Lift. We hypothesize that this difference may be an artifact of training, as verbs such as "lift" are heavily underrepresented in the DROID dataset, which focuses mostly on pick-and-place tasks. Due to the planning nature of the agent and the task specification as an image goal, RoboJEPA models are not susceptible to this type of degradation. Qualitatively, behavior transitioned gradually from smooth but less adaptive execution in smaller models to more reliable recovery from mistakes in larger models (Appendix H.7). Table 3 reports task progress and success rates for the largest RoboJEPA models and VLA references.

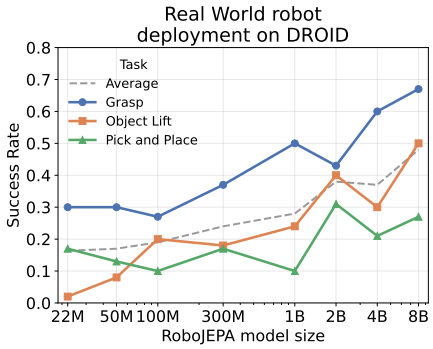


Figure 4: **Real-world deployment on DROID.** Success rates across model sizes for three manipulation tasks; the dashed line shows their average.

Table 3: **Real-robot task progress and success rates (%).** Full results are in Table 12.

Model	Goal	Grasp		Object Lift		Pick and Place	
		Progress	Success	Progress	Success	Progress	Success
π_0-FAST	Text	35	22	20	12	55	40
$\pi_{0.5}$		25	5	12	0	69	53
RoboJEPA-4B	Image	60	60	54	30	49	21
RoboJEPA-8B		67	67	65	50	42	27

4 Conclusion

We introduced RoboJEPA, a family of 22M–8B action-conditioned latent world models trained on large-scale, multi-embodiment robotic data, including the largest JEPA predictor model trained to date. Using this family and the associated training recipe, we established the scaling laws of prediction error in both real and simulated embodiments, demonstrated the correlation between offline prediction error and online downstream planning performance, and showed that the induced policies improved in success rate and qualitative behavior as model scale increased. We will release our data recipe, training and deployment code, model weights, and a diffusion decoder to visualize predicted action-conditioned futures in pixel space. Future directions include (1) jointly scaling the predictor and encoder, (2) expanding the dataset to reduce multi-epoch training and address the data saturation suggested by our fits, (3) using text conditioning to remove the need for image goals, and (4) incorporating learned policy proposals for more efficient planning. These results establish a connection between training compute, prediction quality, and robotic planning performance that can guide the development of more capable world models.

AI USE STATEMENT

In this work, we used AI tools to correct the grammar and punctuation of the main paper text, to structure figures and tables, and to fix latex compilation errors. We did not use any AI tools to assemble and unify open robotic datasets. During the work on RoboJEPA we developed the architecture of the model, unified multiembodiment dataset and dataloader, the KV-cache inference, the multi-GPU planning algorithm and infrastructure, the offline data evaluations without the use of any AI-tools. In the later stage of the project, we used AI tools to implement various distributed training strategies (e.g. tensor-parallel or FSDP training), to build the inference complilation pipeline that allowed us to speedup planning, to create custom environments in RoboCasa, to refactor code before the code release preparation. Additionally, we used AI tools to draft the description of the technical details in several sections of appendix based on the code of RoboJEPA and the main paper text. We manually verified all AI-written text in the appendix for correctness. We used AI tools to assist literature search and before submitting the work we verified the correctness of all citations manually. We take responsibility for the final content of this work, including the text, figures and numerical results.

ETHICS STATEMENT

This work uses previously collected robotic data and simulated robotic environments; it does not involve human subjects or collection of personally identifiable information. We describe our data processing and evaluation procedures in this paper to support responsible use and reproducibility. As with any robotic control system, deployment should be performed with appropriate hardware safeguards and supervision.

REPRODUCIBILITY STATEMENT

We provide the training recipe, model architecture, data mixture, evaluation protocols, scaling-law fitting procedure, and planning configurations in the main paper and appendices. The paper also specifies the checkpoint scales, compute accounting, and figure-generation procedures used for the reported results. We will release the training code with the final version of the paper, together with the model checkpoints and robot-deployment code, to support reproduction of the experiments.

REFERENCES

1X Technologies. 1X world model dataset. https://huggingface.co/datasets/1x-technologies/world_model_raw_data, 2024.

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Bo Ai, Liu Dai, et al. Towards embodiment scaling laws in robot locomotion. In *Conference on Robot Learning (CoRL)*, 2025.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2024. doi: 10.1145/3620665.3640366.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: DINO as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025. URL https://arxiv.org/abs/2507.19468.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. In *Transactions on Machine Learning Research (TMLR)*, 2024.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. RoboAgent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Florian Bordes, Quentin Garrido, Justine T Kao, Adina Williams, Michael Rabbat, and Emmanuel Dupoux. IntPhys 2: Benchmarking intuitive physics understanding in complex synthetic environments. *arXiv preprint arXiv:2506.09849*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. *arXiv preprint arXiv:2210.14891*, 2022.

Ethan Caballero, Priyank Jaini, David Krueger, and Irina Rish. Unified neural scaling laws. *arXiv preprint arXiv:2605.26248*, 2026.

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Khalil Meftah, Maxime Ellerbach, Jess Moss, and Thomas Wolf. LeRobot: State-of-the-art machine learning for real-world robotics in PyTorch. `https://github.com/huggingface/lerobot`, 2024.

Xiaowei Cai, Yunuo Cai, Bingao Chen, Jingxiao Chen, Zhi Chen, Siyuan Feng, et al. τ_0-VLA: a hierarchical robot foundation model with world-model-guided test-time computation. *arXiv preprint arXiv:2608.16885*, 2026.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 24081–24125, 2024a. doi: `10.52202/079017-0759`. URL `https://proceedings.neurips.cc/paper_files/paper/2024/hash/2aee1c4159e48407d68fe16ae8e6e49e-Abstract-Conference.html`.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024b.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. RoboNet: Large-scale multi-robot learning. In *Conference on Robot Learning (CoRL)*, 2019.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Dyna Robotics. Dyna-2: A 1-million-hour scaling law for world-action models. `https://www.dyna.co/dyna-2`, 2026. Accessed: 2026-09-04.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. In *Robotics: Science and Systems (RSS)*, 2022.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: `10.15607/RSS.2024.XX.090`. URL `https://www.roboticsproceedings.org/rss20/p090.html`.

Google DeepMind. Genie 2: A large-scale foundation world model. `https://deepmind.google/blog/genie-2-a-large-scale-foundation-world-model/`, 2024. Technical report / blog post.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations (ICLR)*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 30016–30030, 2022. doi: 10.52202/068431-2176. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract.html.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning (ICML)*, 2023.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, et al. PointWorld: Scaling 3d world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 651–673, 2018. URL https://proceedings.mlr.press/v87/kalashnikov18a.html.

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. MT-Opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Yann LeCun. A path towards autonomous machine intelligence. Open Review, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf.

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2025.

Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: A functional manipulation benchmark for generalizable robotic learning. *The International Journal of Robotics Research*, 2024.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. In *IEEE Robotics and Automation Letters (RA-L)*, 2023.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. doi: 10.1016/S0005-1098(99)00214-9.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations (ICLR)*, 2023.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. *arXiv preprint arXiv:2603.04356*, 2026.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning (ICML)*, 2021.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or semantics? what makes a latent space useful for robotic world models. *arXiv preprint arXiv:2605.06388*, 2026. URL https://arxiv.org/abs/2605.06388.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.

PAN Team, Jiannan Xiang, Yi Gu, Zihan Liu, Zeyu Feng, Qiyue Gao, Yiyan Hu, Benhao Huang, Guangyi Liu, Yichi Yang, Kun Zhou, et al. PAN: A world model for general, interactable, and long-horizon world simulation. *arXiv preprint arXiv:2511.09057*, 2025. URL https://arxiv.org/abs/2511.09057v1.

Tim Pearce and Jinyeop Song. Reconciling kaplan and chinchilla scaling laws. *arXiv preprint arXiv:2406.12907*, 2024.

Tim Pearce, Tabish Rashid, Dave Bignell, Raluca Georgescu, Sam Devlin, and Katja Hofmann. Scaling laws for pre-training agents and world models. *arXiv preprint arXiv:2411.04434*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. *arXiv preprint arXiv:2505.20171*, 2025.

Samuele Punzo, Niccolò Caselli, Ippokratis Pantelidis, Francesco Massafra, Salvatore Lo Sardo, and Mohammadreza Salehi. How do video foundation models encode intuitive physics? probing across pretraining paradigms. *arXiv preprint arXiv:2606.09646*, 2026.

Reuven Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Sebastian Sartor and Neil C. Thompson. Neural scaling laws in robotics. *arXiv preprint arXiv:2405.14005*, 2024.

Makoto Sato, Ryosuke Unno, Masahiro Negishi, Koudai Tabata, Taiju Watanabe, Junnosuke Kamohara, Taiga Kume, Ryo Okada, Yusuke Iwasawa, and Yutaka Matsuo. Scaling laws of model size for world models. In *Proceedings of the Annual Conference of the Japanese Society for Artificial Intelligence (JSAI)*, 2023. doi: 10.11517/pjsai.JSAI2023.0_2G5OS21e02. In Japanese.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning (CoRL)*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=cHZn5Gdh8e.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Wayve. GAIA-3: Scaling world models to power safety and evaluation. https://wayve.ai/thinking/gaia-3/, 2025. Accessed: 2026-09-04.

Wayve. GAIA-4: Multimodal world models powering closed-loop simulation for safe and scalable autonomy. https://wayve.ai/thinking/gaia-4/, 2026. Accessed: 2026-09-04.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. doi: 10.1109/ICRA.2017.7989202.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are scalable world models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, Zhen Zhao, Guangyu Li, Zhao Jin, Lecheng Wang, Jilei Mao, Xinhua Wang, Shichao Fan, Ning Liu, Pei Ren, Qiang Zhang, Yaoxu Lyu, Mengzhen Liu, Jingyang He, Yulin Luo, Zeyu Gao, Chenxuan Li, Chenyang Gu, Yankai Fu, Di Wu, Xingyu Wang, Sixiang Chen, Zhenyu Wang, Pengju An, Siyuan Qian, Shanghang Zhang, and Jian Tang. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024b. URL https://arxiv.org/abs/2412.13877v1.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. DayDreamer: World models for physical robot learning. In *Conference on Robot Learning (CoRL)*, 2023.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, pp. 45210–45234, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c4d66eae503694424123b93ac0fbaf17-Abstract-Conference.html.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, et al. Hierarchical planning with latent world models. *arXiv preprint arXiv:2604.03208*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.