LEARNING MODULAR WORLD MODELS BY COMPOSING REUSABLE DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Massively multitask world models promise to turn broad interaction data into reusable dynamics for control, but such reuse is not guaranteed by pooling tasks into a single dense conditional model. In novel tasks, a held-out environment may combine familiar objects, goals, initial states, or interaction patterns in a configuration never observed during training. We formulate this challenge as *concept accumulation and recombination*: each environment constrains only part of a latent concept library, while evaluation requires composing the learned parts under new task contexts. As a guiding analysis, we give an idealized condition for selecting predictive sources on observed regimes and distinguish this result from the additional structure needed for unseen recombinations. Motivated by this principle, we introduce CWM, a compositional task-routing and dynamics-masking framework that separates task-expert routing from entity and interaction masking, selecting inputs to shared modules that model how they evolve under actions. This simple structural bias encourages the world model to share reusable transition rules while suppressing task-irrelevant factors and environment-specific co-occurrence shortcuts. On MM-Bench, which spans more than 200 tasks across 10 domains, CWM preserves the scalability of massively multitask training, yielding around a 32% overall gain over the state-of-the-art. It also provides stronger adaptation to held-out compositional tasks, showing that a sparse task-routing and dynamics-masking inductive bias can improve both large-scale multitask learning and generalization to new task recombinations.

1 INTRODUCTION

World models have become a practical foundation for reinforcement learning and control (Sutton, 1991; Ha and Schmidhuber, 2018; Hafner et al., 2020; Schrittwieser et al., 2020; Ye et al., 2021; Hafner et al., 2021; 2025; Guo et al., 2025; Ye et al., 2026). By learning an action-conditioned model of the environment, an agent can imagine future trajectories and use those predictions for planning (Hansen et al., 2022; Zhou et al., 2024), policy evaluation (Quevedo et al., 2025), or policy improvement (Cen et al., 2025; Sharma et al., 2026). Recent work has begun to scale this idea from single-task control to pretrained and massively multitask world models, trained on broad collections of demonstrations, interaction data, or task-conditioned experience (Georgiev et al., 2024; Zhang et al., 2025; Nicklas Hansen, 2026). A central promise is that a sufficiently broad world model should not merely fit many tasks, but should turn experience from those tasks into reusable dynamics for future control.

Many control tasks are naturally compositional: new tasks often combine familiar objects, skills, dynamics, or action effects in configurations that were not observed during training, *e.g.*, an agent may have seen how to push, lift one object, or navigate around another object, but a new task may require these components to be composed differently. Thus, a key factor for large-scale world models is whether they can generalize compositionally to *unseen tasks* of previously learned factors.

Most existing multitask or pretrained world-model methods treat tasks and environments as data from a shared training pool, and rely on a single conditional model to absorb their variations (Bruce et al., 2024; Georgiev et al., 2024; Team et al., 2025; Chi et al., 2025; Nicklas Hansen, 2026; Gao et al., 2026). However, scaling by pooling more tasks and environments does not by itself guarantee this level of generalization. This design can work well when test tasks resemble seen task distributions,

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.

but it provides little explicit structure for deciding *which parts of the model should be shared*, *which should remain task- or environment-specific*, and *how these parts should be recombined when facing novel task compositions*. As a result, the learned model may entangle reusable dynamics, task semantics, and environment-specific factors, limiting its ability to systematically generalize to new combinations of familiar components. This creates a natural learning objective that differs from standard single-environment model learning: *a world model should accumulate reusable concepts across environments and compose them to handle a new environment that recombines these concepts.*

This objective suggests a simple principle for scalable world-model learning: the model should separate *which* latent concepts and relations are predictively relevant from *how* the selected concepts interact. Following this principle, we propose CWM, a compositional framework that treats multitask world-model learning as concept accumulation and recombination.

Accumulate	**Mask**	**Compose**
Learn reusable concepts from partial environments.	Use sparse masks to select predictive inputs.	Recombine concepts through a shared interaction module.

We motivate source selection through an idealized sparse latent-dynamics proposition for given partitions and observed regimes. Support recovery alone does not determine an unseen interaction; an appendix illustration makes explicit the additional mechanism structure and training coverage that can permit recombination. These results motivate the design, rather than guarantee recovery or generalization by the end-to-end learner. In practice, CWM partitions the latent state into concept-level components, learns task routing with amortized entity and interaction masks, and composes selected components through a shared interaction module.

We build CWM on the NewT (Nicklas Hansen, 2026) backbone, preserving its scalability while adding an explicit mechanism for compositional generalization. We evaluate CWM on MM-Bench (Nicklas Hansen, 2026), a large-scale benchmark covering more than 200 embodied control tasks across 10 domains. Our evaluation includes both in-distribution multitask learning and adaptation to held-out compositional tasks, whose factors were observed during source training but not jointly seen in the same task configuration. Across domains, CWM improves over NewT and other multitask model-based and model-free baselines, while preserving the scalability of large-scale joint training. The gains are especially clear during adaptation to held-out compositional tasks, where CWM achieves stronger and more stable performance under unseen task compositions. Ablations further show that task routing and the learned masks are complementary: task routing, entity masking, and interaction masking each contribute to generalization, supporting our hypothesis that structured routing and masking help the world model accumulate and recombine concepts at test time. See §A for more related work.

2 Latent Concept Recombination in Multi-Environment World Models

We formalize the target structure for multi-environment world-model learning. The key distinction is between reusable concepts across a task family and the subset that is predictively relevant in a particular environment or transition regime. Each environment constrains only part of this latent concept library, while evaluation may require familiar concepts and interactions in a new combination; the notation below makes this recombination target precise.

Multi-environment POMDPs. We learn a single world model from data collected across multiple environments. Let $\mathcal{E}_{\text{tr}} = \{1, \dots, N\}$ denote the set of training environments. Each environment $e \in \mathcal{E}_{\text{tr}}$ induces a POMDP $\mathcal{M}_e = (\mathcal{S}_e, \mathcal{A}_e, \mathcal{Y}_e, P_e, O_e, R_e, \rho_e; \mathbf{g}_e)$, where $\mathcal{S}_e$ is the latent environment state, $\mathcal{A}_e$ is the action space, $\mathcal{Y}_e$ is the observation space, P_e is the transition kernel, O_e is the observation model, R_e is the reward function, ρ_e is the initial-state distribution, and $\mathbf{g}_e$ is a task context such as a language instruction or task embedding. Across environments, the transition kernels and rewards may differ, but they are assumed to share a reusable structure that can be represented in a unified latent space. We express the intended structure through latent partitions and source-target dynamics links. The analysis takes these partitions as given; it does not establish that the learned representation recovers underlying semantic factors.

Concept partitions and predictive relevance. We write the learned latent state as K components:

$$\mathbf{z}_t = [\mathbf{z}_{1,t}, \ldots, \mathbf{z}_{K,t}], \qquad \mathbf{z}_{i,t} \in \mathcal{Z}_i. \tag{1}$$

We call each $\mathbf{z}_{i,t}$ a *concept partition*. A partition may capture a reusable latent factor, such as an entity slot, object attribute, gripper state, tool state, or contact-related variable. It need not be human-labeled, and the important property is that its dynamics can be consistently reused across environments. These partitions remain part of $\mathbf{z}_t$ in every environment; the paper does not mask the latent state itself. A partition may be constant, nuisance, or predictively irrelevant on a particular support. For example, one manipulation environment may vary a drawer and block, while another may vary a tool and block. Each environment constrains the partitions and interactions it uses and leaves the others weakly constrained. A held-out task can then ask the robot to use the drawer, tool, block, gripper, and familiar contact effects together under a new task context $\mathbf{g}_\star$. The individual concepts are not new; their joint predictive use is. §3 refines this intuition to transition regimes, where predictive relevance can vary with state, action, and task context.

Sparse latent dynamics. Partition relevance summarizes which factors can be constrained; recombination also requires a rule for how different source partitions affect each target partition under actions. The setting analyzed in §3 assumes the following sparse parametric compositional form: Let $c_t = \kappa(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}_e)$ denote the current transition regime. Let $\mathcal{C}_{\mathrm{tr}}$ collect the transition regimes observed in training. For each regime c, let $\mathbf{M}^\star_c \in \{0,1\}^{K\times K}$, where $M^\star_{c,ki} = 1$ means partition i is predictively relevant for updating partition k in regime c. For each target partition k, define the row-wise dynamics support as $\mathcal{I}^{(k)}_c = \{i : M^\star_{c,ki} = 1\}$.

$$\mathbf{z}_{k,t+1} = g^{(k)}_{\theta^\star}\left(\mathbf{a}_t,\ [M^\star_{c_t,k1} f^\star_1(\mathbf{z}_{1,t}), \ldots, M^\star_{c_t,kK} f^\star_K(\mathbf{z}_{K,t})]\right), \qquad k \in [K], \tag{2}$$

where $f^\star_i : \mathcal{Z}_i \to \mathcal{H}_i = \mathbb{R}^{d_i}$ returns a vector feature for partition i, $M^\star_{c_t,ki}$ is a scalar gate, and $g^{(k)}_{\theta^\star}$ is the k-th output head of one shared interaction rule. Equation (2) defines an idealized deterministic data-generating process for the support analysis, not a claim that the full implementation has this form. Different regimes use different source-target supports, while a shared interaction rule composes the selected factors with the action. The motivating preference is for complex local transformations but a stable, simple interaction rule; the support proposition alone does not formalize or guarantee this latter property. In the running example, drawer-block and tool-block data can separately constrain object persistence, controllability, contact response, and relative geometry. A held-out drawer-tool-block task is difficult because the joint predictive use was not observed. Stability of local regularities helps motivate reuse, but is not sufficient without restrictions on their combination and informative training coverage. If contact effects or object persistence changed freely whenever the tool and drawer co-occurred, no model could infer recombination without observing it directly.

Goal: Accumulate and compose. After training on environments whose observed transition regimes form $\mathcal{C}_{\mathrm{tr}}$, we want the world model to generalize to a new environment $\mathcal{E}_\star$ whose row-wise dynamics supports recombine observed source-target links:

$$\mathcal{I}^{(k)}_\star \subseteq \bigcup_{c\in\mathcal{C}_{\mathrm{tr}}} \mathcal{I}^{(k)}_c, \qquad k \in [K], \tag{3}$$

where $\mathcal{I}^{(k)}_\star = \{i : M^\star_{\star,ki} = 1\}$ is the held-out source support for target partition k. The challenge is that familiar concepts and action effects appear together in a joint configuration absent from training. Fig. 1 illustrates this view: for partition $\mathbf{z}_{3,t+1}$, training regimes expose links involving $\{1,3\}$ or $\{2,3\}$, while evaluation selects familiar links when $\{1,2,3\}$ participate together. The learning target is therefore to select useful source-target links, accumulate reusable dynamics from partial environments, and reuse them in held-out environments. Coverage of individual links does not by itself identify their joint effect. The next section isolates what sparse support selection can establish and what further conditions recombination would require.

3 An Idealized Motivation for Sparse Predictive Masking

We isolate a limited question: when can sparse masking select the sources needed for prediction on observed regimes? The analysis takes the latent partitions and transition regimes as given; it does not establish that they are discovered by the encoder. We then distinguish this support-selection question from predicting an unseen combination of sources.

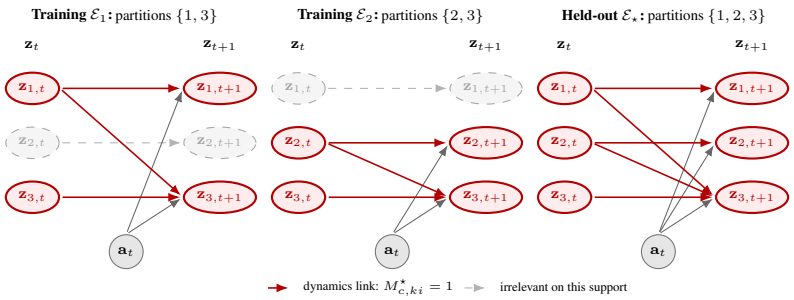

Figure 1: **Knowledge accumulation and composition.** Nodes show concept partitions; red nodes are constrained in the current scenario. Red arrows show selected source-target dynamics links used by (2); gray dashed partitions remain present in $\mathbf{z}_t$ but are weakly constrained or predictively irrelevant on that support. The held-out scenario illustrates the goal of reusing familiar links in a new combination, not a consequence of link coverage alone.

Idealized masked predictor. Let $c_t = \kappa(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}_e)$ index one of finitely many training regimes, with transition distribution $\mathcal{D}_c$. Each source is encoded locally as $\mathbf{h}_{i,t} = f_i(\mathbf{z}_{i,t}) \in \mathcal{H}_i$, and a fixed binary mask for that regime selects the inputs to each target update:

$$\hat{\mathbf{z}}_{k,t+1} = g_\theta^{(k)}(\mathbf{a}_t, [M_{c_t,k1}\mathbf{h}_{1,t}, \ldots, M_{c_t,kK}\mathbf{h}_{K,t}]), \qquad \mathbf{M}_c \in \{0,1\}^{K\times K}. \tag{4}$$

Within a fixed regime, an omitted source has no other path to the target prediction. Assume square-integrable deterministic dynamics of the form in (2), realizable by this predictor class with the oracle masks $\{\mathbf{M}_c^\star\}$. Define the population fit loss and ideal support-selection problem by

$$\mathcal{L}_{\text{fit}} = \sum_{c\in\mathcal{C}_{\text{tr}}} \mathbb{E}_{\mathcal{D}_c}\left[\sum_{k=1}^{K} \|\hat{\mathbf{z}}_{k,t+1} - \mathbf{z}_{k,t+1}\|^2\right], \tag{5}$$

$$\min_{\theta,\{f_i\},\{\mathbf{M}_c\}} \sum_{c\in\mathcal{C}_{\text{tr}}} \|\mathbf{M}_c\|_0 \quad \text{subject to} \quad \mathcal{L}_{\text{fit}} = 0. \tag{6}$$

This constrained population problem isolates the role of sparsity; it is not the finite-sample neural training objective.

Assumption 3.1 (Predictive relevance on observed regimes)**.** *For each training regime c, target k, and required source $i \in \mathcal{I}_c^{(k)}$, omitting that source leaves positive prediction error even when all remaining inputs are available:*

$$\Delta_{c,k,i} := \inf_v \mathbb{E}_{\mathcal{D}_c}\left[\|\mathbf{z}_{k,t+1} - v(\mathbf{z}_{-i,t}, \mathbf{a}_t, \mathbf{g}_e)\|^2\right] > 0, \tag{7}$$

where the infimum is over measurable predictors with finite squared risk.

This condition requires the training distribution to reveal information unique to each required source. For example, perfectly duplicated sources would not be separately identifiable. It concerns prediction on observed regimes, not behavior on a held-out combination.

Proposition 3.2 (Predictive support on observed regimes)**.** *Under the realizability and hard-masking conditions above and Assumption 3.1, every solution of (6) satisfies $\mathbf{M}_c = \mathbf{M}_c^\star$ for all training regimes c.*

Proof idea. Omitting a required source incurs error at least $\Delta_{c,k,i} > 0$, so every exact-fit predictor includes every required link. The realizable oracle uses precisely these links; minimum support therefore leaves no extra links. The full proof is in Appendix B. The proposition does not identify latent coordinates, task experts, or masks for unseen regimes.

Support recovery does not determine recombination. Suppose training contains only $(u, 0)$ and $(0, v)$. Every member of the family

$$F_\gamma(u, v) = u + v + \gamma uv \tag{8}$$

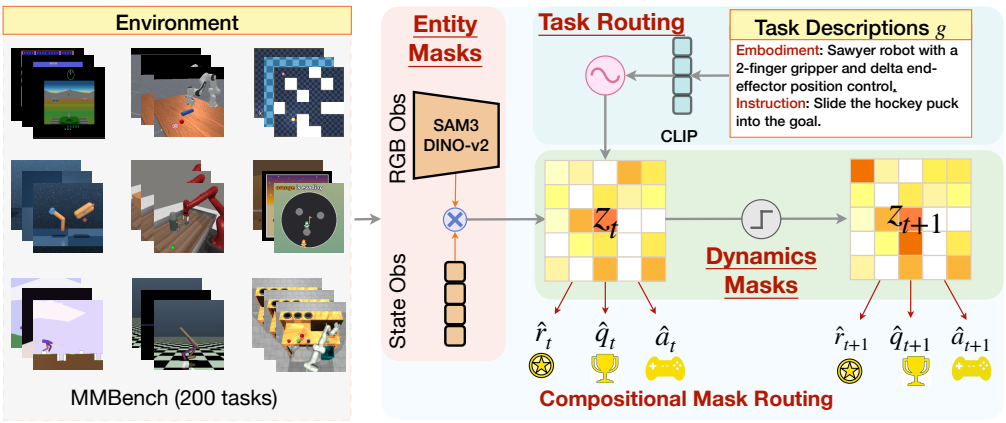

Figure 2: **Overview of compositional task routing and dynamics masking.** Given RGB and state observations, entity masks are extracted with SAM3 and encoded with DINO-v2 to form structured entity-centric latent states. A task router uses the language embedding to select reusable task experts, while dynamics masks select the entities and interactions relevant for the current transition. The resulting latent dynamics predict the next latent state, reward, value, and action prior, enabling the world model to share components across tasks while recombining them under new task compositions.

fits the same training responses and has the same observed source supports, yet the members disagree when both sources are active. Thus partitioning alone does not determine an unseen interaction or increase expressive power over an unrestricted predictor. Its intended benefit is to expose reusable structure on which further inductive biases can act. Appendix B.2 gives one explicit sufficient case: shared bounded-order mechanisms can be recombined when their calibration configurations are covered by training and the target mechanism activations are supplied. This illustration derives its conclusion from structural and coverage conditions, not from an assumption that a complexity criterion selects the correct extension.

Design Motivation

The proposition motivates selecting predictive sources; shared dynamics and sparse routing additionally express a preference for reusable computation. CWM implements these ideas with learned slots, soft gates, and shared neural modules, not with the constrained hard-mask problem above. Its global task features and nonlinear update allow information paths and interactions outside that idealization. The analysis is therefore design motivation, not a guarantee of oracle-mask recovery or held-out prediction by the end-to-end learner. The practical contribution is assessed through the empirical evaluations.

4 Method: Compositional Task Routing and Dynamics Masking

Motivated by the source-selection analysis in §3, we introduce sparse routing and masking into a scalable multitask world model. Our method builds on the NewT backbone (Nicklas Hansen, 2026) with TD-MPC2 (Hansen et al., 2024): given a learned latent state $\mathbf{z}_t$, action $\mathbf{a}_t$, and task context $\mathbf{g}_e$, the model predicts the next latent state, reward, and value targets for planning and actor–critic learning. Specifically, we modify its latent transition with compositional task routing and dynamics masking, so the model can share task-conditioned computation through reusable experts while focusing on predictively relevant concepts and interactions.

4.1 Backbone: Multitask World Model (NewT)

We build on the NewT (Nicklas Hansen, 2026), a multitask world model, which extends TD-MPC2 to language-conditioned and optionally image-conditioned control. Given a task context $\mathbf{g}$ from a language encoder, an optional image embedding $\mathbf{x}$, and a low-dimensional state observation $\mathbf{s}_{\text{state}}$, the encoder maps the current observation into a latent state $\mathbf{z}_t = e(\mathbf{s}_{\text{state},t}, \mathbf{x}_t, \mathbf{g})$. It then predicts the

next latent state, reward, value target, and policy prior:

$$\mathbf{z}'_{t+1} = d(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}), \qquad \hat{r}_t = R(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}), \qquad \hat{q}_t = Q(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}), \qquad \hat{\mathbf{a}}_t = \pi(\mathbf{z}_t, \mathbf{g}). \tag{9}$$

Unlike generative world models that decode future observations, this self-predictive objective trains the latent dynamics to be control-centric. NewT jointly trains the encoder, dynamics, reward, and value heads through a one-step latent prediction objective. Given a transition $(\mathbf{s}_{\text{state},t}, \mathbf{x}_t, a_t, \mathbf{s}_{\text{state},t+1}, \mathbf{x}_{t+1}, \mathbf{g})$, we define

$$\ell^{(t)}_{\text{wm}}(\theta) = \|d_\theta(\mathbf{z}_t, a_t, \mathbf{g}) - \mathtt{sg}(e_\theta(\mathbf{s}_{\text{state},t+1}, \mathbf{x}_{t+1}, \mathbf{g}))\|_2^2 + \ell_{\text{CE}}(\hat{r}_t, r_t) + \ell_{\text{CE}}(\hat{q}_t, q_t), \tag{10}$$

where $\mathtt{sg}$ denotes stop-gradient, ℓ_{CE} is the discrete regression loss used for rewards and values, and d_θ denotes the latent dynamics model.

The full training objective is then $\mathcal{L}_{\text{wm}}(\theta) = \mathbb{E}_{\tau\sim\mathcal{B}}\left[\sum_{t=0}^{H} \lambda^t \ell^{(t)}_{\text{wm}}(\theta)\right]$. where λ downweights temporally distant rollout steps. The value target follows the TD-MPC2 update, $q_t = r_t + \gamma Q_{\text{tgt}}(\mathbf{z}'_{t+1}, p(\mathbf{z}'_{t+1}, \mathbf{g}), \mathbf{g})$, where Q_{tgt} is an EMA target network.

Our method keeps this training pipeline, with replay structure, reward and value heads, policy objective, and planning procedure unchanged from NewT. Appendix D gives backbone details. NewT provides a scalable objective for massively multitask control, but its standard transition model treats task variation through a single conditional dynamics function. Below, we replace $d(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g})$ with a task-routed, mask-conditioned transition that exposes predictive latent components and interactions.

4.2 Compositional Task Routing and Dynamics Masking

We replace the monolithic NewT transition $d(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g})$ with a task-routed, mask-conditioned transition model that masks and composes task-relevant latent components (Fig. 2). The sparse objective in §3 treats the regime support matrix $\mathbf{M}_c$ as an explicit binary variable for analysis. In the implementation, we do not introduce separate mask parameters for each environment. Instead, the masks are amortized: mask networks predict transition-dependent soft masks from the current latent state, action, and task context. Thus the empirical mask vectors $\mathbf{m}^{\text{ent}}_t$ and $\mathbf{m}^{\text{int}}_t$ are induced by shared mask networks rather than stored as free per-environment variables. Let ϑ collect all learnable parameters of the encoder, backbone heads, local dynamics maps, task experts, shared interaction maps, task-routing networks, and mask networks. The analysis assumes given partitions and fixed regime masks in order to isolate source exclusion. The implementation instead uses entity-centric slots, task-level mixture routing, and transition-dependent soft gates. These gates control selected computational paths, but global task features and mask networks can also carry information across slots. They are therefore not established estimates of the complete oracle supports $\mathbf{M}^\star_{c_t}$ in §3.

Entity-level latent masks. We implement the concept partitions by organizing the NewT latent state into entity-centric slots. For each observation, SAM-3 (Carion et al., 2025) produces object or region masks, and a DINO-v2 encoder (Oquab et al., 2023) maps each masked region into an entity representation: $\mathbf{z}_t = \{\mathbf{z}^{(n)}_t\}_{n=1}^{N_{\text{ent}}}$, where N_{ent} is the number of latent entity slots.

Task-level mixture routing. NewT conditions its transition model on the task embedding $\mathbf{g}$ by concatenating $\mathbf{g}$ with the latent state and action. We instead use $\mathbf{g}$ as a sparse mixture-of-experts (Sparse MoE) router (Shazeer et al., 2017). Given $\mathbf{g}$, the router predicts mixture weights over J task experts:

$$\boldsymbol{\alpha}^{\text{task}}(\mathbf{g}) = \text{TopK}\big(\text{softmax}\left(r^{\text{task}}_\psi(\mathbf{g})/\tau\right)\big) \in \Delta^J, \tag{11}$$

where only the top-K_{task} experts are kept and renormalized and $\Delta^J = \{\boldsymbol{\alpha} \in \mathbb{R}^J_{\geq 0} : \sum_{j=1}^J \alpha_j = 1\}$ denotes the J-dimensional probability simplex. Let $\mathbf{h}_t$ denote the flattened structured latent representation from all entity groups. Each task expert produces a task-specific transition feature, $\mathbf{u}^{\text{task}}_{j,t} = E^{\text{task}}_j(\mathbf{h}_t, \mathbf{a}_t, \mathbf{g})$, and the routed task feature is $\mathbf{u}^{\text{task}}_t = \sum_{j=1}^J \alpha^{\text{task}}_j(\mathbf{g})\, \mathbf{u}^{\text{task}}_{j,t}$. This makes task conditioning as a sparse composition of reusable task experts, allowing related tasks to share experts while still specializing to different action effects or task families.

Dynamic interaction masks. We implement the dynamic interaction mask as a lightweight graph transition over entity slots. Each slot $\mathbf{z}^{(n)}_t$ is treated as a node. A shared local dynamics encoder

produces node features $\mathbf{h}_t^{(n)} = f_\theta^{\text{dyn}}(\mathbf{z}_t^{(n)}, \mathbf{a}_t, \mathbf{u}_t^{\text{task}})$, which are gated by an entity mask $\mathbf{m}_t^{\text{ent}} = \sigma(r_\phi^{\text{ent}}(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g})) \in [0,1]^{N_{\text{ent}}}$:

$$\tilde{\mathbf{h}}_t^{(n)} = m_t^{\text{ent},(n)} \mathbf{h}_t^{(n)}. \tag{12}$$

For cross-entity effects, we compute pairwise messages $\mathbf{v}_t^{(n,\ell)} = \psi_\theta(\mathbf{z}_t^{(n)}, \mathbf{z}_t^{(\ell)}, \mathbf{a}_t, \mathbf{u}_t^{\text{task}})$ and gate them with an interaction mask $\mathbf{m}_t^{\text{int}} = \sigma(r_\phi^{\text{int}}(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g})) \in [0,1]^{N_{\text{ent}} \times N_{\text{ent}}}$. The message received by entity n and its predicted next state are

$$\mathbf{c}_t^{(n)} = \sum_{\ell \neq n} m_t^{\text{int},(n,\ell)} \mathbf{v}_t^{(n,\ell)}, \qquad \hat{\mathbf{z}}_{t+1}^{(n)} = u_\theta\left(\mathbf{z}_t^{(n)}, \tilde{\mathbf{h}}_t^{(n)}, \mathbf{c}_t^{(n)}, \mathbf{a}_t, \mathbf{u}_t^{\text{task}}\right). \tag{13}$$

The entity mask weights local slot features, and the interaction mask weights directed pairwise messages used for next-state prediction. All task-routing and dynamics-masking components are trained end-to-end with the NewT world-model objective. We add ℓ_1 sparsity penalties on the induced entity and interaction masks, together with the loss for the task experts:

$$\mathcal{L}_{\text{wm}}^{\text{route}}(\vartheta) = \mathcal{L}_{\text{wm}}(\vartheta) + \mathbb{E}_{\tau \sim \mathcal{B}}\left[\sum_{t=0}^{H} \lambda^t \left(\beta_{\text{ent}} \|\mathbf{m}_t^{\text{ent}}\|_1 + \beta_{\text{int}} \|\mathbf{m}_t^{\text{int}}\|_1\right)\right] + \beta_{\text{bal}} \mathcal{L}_{\text{bal}}\left(\boldsymbol{\alpha}^{\text{task}}\right). \tag{14}$$

The ℓ_1 terms encourage small soft gates, expressing a practical sparsity preference inspired by (6). They do not solve its exact-fit, minimum-cardinality problem: prediction and regularization are traded off, and gating a message does not exclude every other information path from its source. The shared modules and penalties encourage reusable computation and selective routing; their benefits are evaluated empirically.

4.3 Planning and Policy Learning

We keep the planning and policy-learning procedure of the NewT backbone unchanged. During environment interaction, actions are selected by model predictive control using the learned world model, with the policy prior $\pi(\mathbf{z}_t, \mathbf{g})$ used to warm-start candidate action sequences. Hence, the only difference is that imagined rollouts use our task-routed, mask-conditioned transition $\hat{\mathbf{z}}_{t+1} = d_{\text{route}}(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g})$ instead of the standard monolithic transition $d(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g})$. The policy prior is trained with the same model-based behavior cloning, value maximization, and entropy terms as in NewT. Thus, improvements in control arise purely from a more compositional latent dynamics model.

5 Experiments

We evaluate whether compositional task routing and dynamics masking improve large-scale world-model learning in two complementary settings. The first is standard multitask learning on MM-Bench, where the goal is to study whether the added structure *preserves the scalability* of NewT while improving performance across a heterogeneous 200-task training distribution. The second is held-out compositional generalization, where evaluation tasks are formed by recombining factors that were observed during training but never jointly encountered, to evaluate the *generalization capability* empowered by task routing and dynamics masking. Together, these settings test both sides of our hypothesis: sparse task routing and dynamics masking should improve multitask fitting while producing a world model whose learned components can be reused under new task compositions.

Hence, the experiments are organized around three questions.
(Q1) Scalable multitask learning: Does CWM improve in-distribution performance across many domains without sacrificing the large-scale training stability of the NewT backbone?
(Q2) Held-out compositional adaptation: After the same 200-task multitask training stage, does CWM provide a better initialization for unseen task compositions?
(Q3) Routing and masking component analysis: Which parts of the design matter most, and do task routing and entity- and interaction-level masks provide complementary benefits?

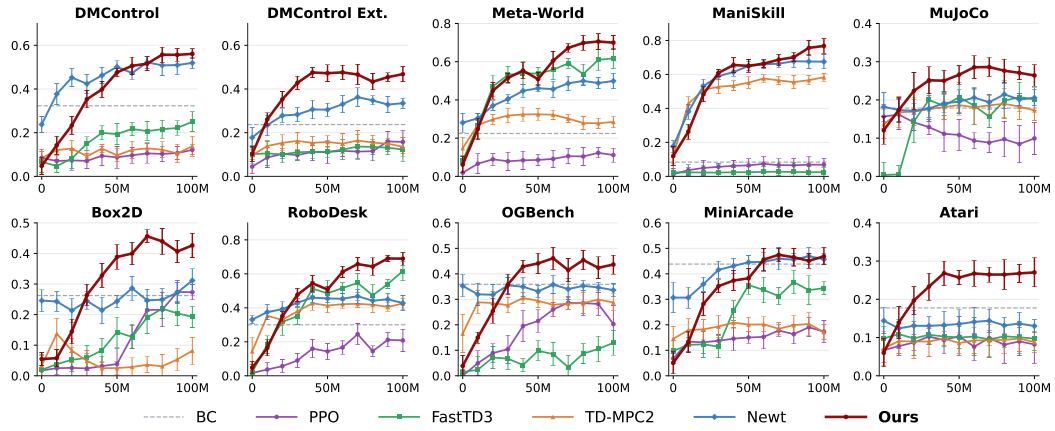

Figure 3: **Average score across domains.** The curves compare CWM with baselines on evaluation tasks formed by recombining factors observed during training but not jointly observed. Standard errors are calculated with 3 random seeds.

5.1 Experimental Setup

We evaluate CWM on MM-Bench (Nicklas Hansen, 2026), a large-scale benchmark suite for embodied decision-making and generalization. MM-Bench contains more than 200 tasks across 10 domains: DMControl (Tassa et al., 2018), DMControl Extended, Meta-World (Yu et al., 2020), ManiSkill3 (Tao et al., 2024), MuJoCo (Todorov et al., 2012), Box2D (Brockman et al., 2016), RoboDesk (Kannan et al., 2021), OGBench (Park et al., 2024), MiniArcade (Nicklas Hansen, 2026), and Atari (Bellemare et al., 2013). The tasks span locomotion, manipulation, contact-rich interaction, navigation, and long-horizon game-like control, making MM-Bench a suitable testbed for evaluating both scalable multitask learning and compositional generalization.

Evaluation protocol. We evaluate under two settings. In-distribution evaluation measures performance on held-out episodes from the multitask training distribution. For held-out compositional adaptation, target tasks recombine factors observed during pretraining into previously unseen configurations. No trajectories from these target configurations are used during pretraining. After the same 200-task pretraining stage, each compared pretrained model is fine-tuned on each target task with the same 100K-step interaction budget. These results measure adaptation to unseen task compositions.

Baselines. We compare CWM with baselines spanning imitation learning, model-free RL, model-based RL, and multitask world-model learning. These include single-task BC, PPO (Schulman et al., 2017), FastTD3 (Seo et al., 2025), multitask and single-task TD-MPC2 (Hansen et al., 2024) as a task-specific upper reference, and NewT as the strongest massively multitask world-model baseline. The comparison with NewT directly isolates the effect of replacing its dense latent transition with our sparse compositional transition. All methods use the same total interaction budget and evaluation episodes. Multi-task methods train a single agent across all tasks, while single-task baselines train one agent per task with the same per-task budget. Scores are normalized within each domain and reported as domain averages, followed by an aggregate average. Implementation details are in Appendix E.

5.2 Results and Analysis

Multitask training performance. Fig. 3 reports multitask performance across the 10 MM-Bench domains; full quantitative results, including single-task methods, are provided in Appendix Table A2.

Among multi-task methods, CWM achieves the best performance in almost every domain. This indicates that compositional task routing and dynamics masking do not trade off standard multitask performance for generalization; instead, they consistently strengthen the NewT backbone across heterogeneous domains. The gains are especially clear in domains that involve object-centric interactions or compositional task structure, such as Meta-World, ManiSkill, RoboDesk, OGBench, and Atari. These results suggest that explicitly modeling which entities, task factors, and interactions are relevant can help the agent avoid learning overly dense task representations. Instead of sharing all

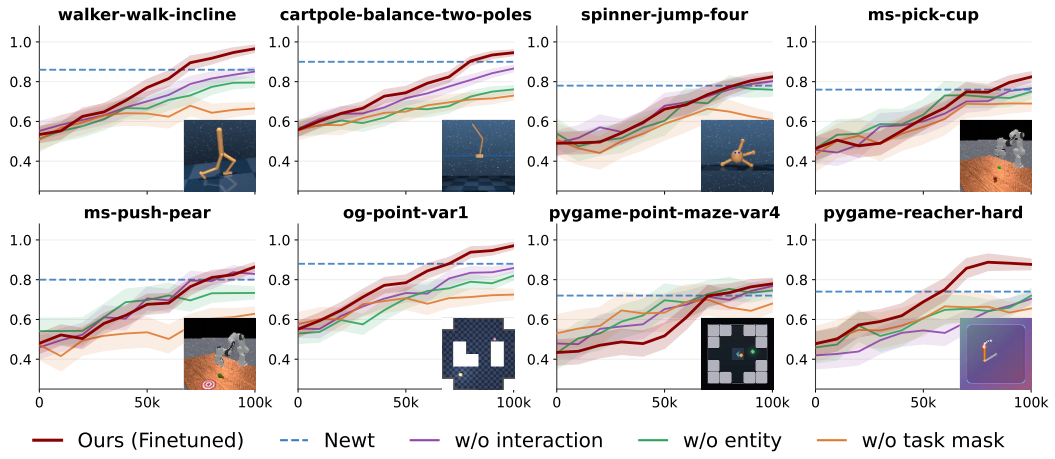

Figure 4: **Compositional adaptation to unseen tasks after multitask training.** We finetuned on held-out compositional tasks whose factors were observed during training but not in the same task configuration. Standard errors are calculated across 3 random seeds.

information uniformly across tasks, the masks provide a structured inductive bias that encourages the world model and policy to reuse the relevant components while suppressing nuisance or task-irrelevant factors. We further ablate task routing and entity- and interaction-level masks in Appendix C.3 and Fig. A1, showing that these components provide complementary benefits for multitask learning.

Compositional adaptation to unseen tasks. Fig. 4 evaluates transfer to held-out compositional tasks after the same 200-task multitask training stage. These tasks are unseen during training but recombine familiar embodiments, objects, goals, and interaction patterns. CWM provides a stronger finetuning initialization than NewT and all ablated variants, improving faster and reaching higher final performance, especially on tasks such as `walker-walk-incline`, `spinner-jump-four`, `og-point-var1`, and `pygame-reacher-hard`. This suggests that routing and masks encourage reusable latent components rather than task-specific dynamics.

The ablations further show that the routing and masking components play complementary roles. Removing the interaction mask hurts transfer when new cross-entity or contact relations matter; removing the entity mask weakens focus on relevant objects or state factors; and removing task routing degrades the selection of an appropriate dynamics prior. Overall, these results show improved unseen-task adaptation with routing and masking, consistent with the intended reuse of shared multitask components. Additional controls for entity representations and expert capacity, along with a known-support recombination experiment, are provided in Appendix C.1.

6 Conclusion

We introduced CWM, a compositional task-routing and dynamics-masking framework that makes reuse explicit in massively multitask world models. Rather than asking one dense conditional model to absorb all task variation, CWM uses task routing and entity- and interaction-level masks to select computational paths and compose their outputs through shared dynamics. Our idealized analysis motivates sparse source selection on observed regimes and clarifies the additional structure and coverage needed to determine unseen combinations. Empirically, CWM improves both in-distribution multitask learning and adaptation to held-out compositional tasks on MM-Bench. These results suggest that scalable world models need not only more pooled data, but also structure that helps them accumulate, mask, route, and recombine reusable dynamics.

Limitations. The analysis takes partitions and regimes as given and does not guarantee recovery by the learned encoder or soft gates. Its recombination illustration assumes a bounded-order mechanism family and calibration coverage that the full architecture need not satisfy. The implementation uses a prescribed entity-slot organization, and evaluation is simulation-focused; future work should investigate learned partition structure, task-routing efficiency as the number of domains grows, and real robotic settings.

AI USE STATEMENT

We used generative AI tools to assist with grammar checking, LaTeX formatting and and mathematical arguments checking only. These tools also assisted with checking consistency between the methodological descriptions and experimental reporting. The authors retain responsibility for verifying all AI-assisted content and for the accuracy and integrity of the final manuscript, including its theoretical claims and experimental results.

ETHICS STATEMENT

This work studies modular world models for reinforcement learning and evaluates them in simulated control environments. It does not involve experiments with human participants or deployment on physical robots. The benchmarks, pretrained models, and third-party software used in this work remain subject to their respective licenses and usage terms.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B present the assumptions and proofs for the idealized theoretical analysis. Section 4 describes the architecture, routing and masking mechanisms, and training objective. Section 5 specifies the evaluation settings, while Appendix C provides detailed results, component ablations, and controlled experiments. Appendix D describes the NewT backbone and its learning objectives, and Appendix E documents the implementation, hyperparameters, observation processing, and training and evaluation protocols.

REFERENCES

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

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

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

N Hansen, X Wang, and H Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning, PMLR*, 2022.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation. *arXiv preprint arXiv:2506.00613*, 2025.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Ansh Kumar Sharma, Yixiang Sun, Ninghao Lu, Yunzhe Zhang, Jiarao Liu, and Sherry Yang. World-gymnast: Training robots with reinforcement learning in a world model. *arXiv preprint arXiv:2602.02454*, 2026.

Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. Pwm: Policy learning with multi-task world models. *arXiv preprint arXiv:2407.02466*, 2024.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M Patel, Paul Pu Liang, et al. World-in-world: World models in a closed-loop world. *arXiv preprint arXiv:2510.18135*, 2025.

Xiaolong Wang Nicklas Hansen, Hao Su. Learning massively multitask world models for continuous control. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MPabX9LEds.

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.

GigaWorld Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jiagang Zhu, Kerui Li, Mengyuan Xu, et al. Gigaworld-0: World models as data engine to empower embodied ai. *arXiv preprint arXiv:2511.19861*, 2025.

Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Harini Kannan, Danijar Hafner, Chelsea Finn, and Dumitru Erhan. Robodesk: A multi-task reinforcement learning benchmark, 2021.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv:2410.20092*, 2024.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. Fasttd3: Simple, fast, and capable reinforcement learning for humanoid control. *arXiv preprint arXiv:2505.22642*, 2025.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Uai*, volume 4, pages 162–169, 2004.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10069–10076, 2020.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pages 5639–5650. PMLR, 2020.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pages 3480–3491. PMLR, 2021.

Tongzhou Wang, Simon Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. In *International Conference on Machine Learning*, pages 22591–22612. PMLR, 2022a.

Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 33:3976–3990, 2020.

Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. *arXiv preprint arXiv:2206.13452*, 2022b.

Tianying Ji, Yongyuan Liang, Yan Zeng, Yu Luo, Guowei Xu, Jiawei Guo, Ruijie Zheng, Furong Huang, Fuchun Sun, and Huazhe Xu. Ace: off-policy actor-critic with causality-aware entropy regularization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 21620–21647, 2024.

Yuren Liu, Biwei Huang, Zhengmao Zhu, Honglong Tian, Mingming Gong, Yang Yu, and Kun Zhang. Learning world models with identifiable factorization. *Advances in Neural Information Processing Systems*, 36:31831–31864, 2023.

Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. AdaRL: What, where, and how to adapt in transfer reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=8H5bpVwvt5.

Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial intelligence*, 121(1-2):49–107, 2000.

Hansen Jin Lillemark, Benhao Huang, Fangneng Zhan, Yilun Du, and Thomas Anderson Keller. Flow equivariant world models: Memory for partially observed dynamic environments. *arXiv preprint arXiv:2601.01075*, 2026.

Zeyu Zhang, Danning Li, Ian Reid, and Richard Hartley. Geoworld: Geometric world models. *arXiv preprint arXiv:2602.23058*, 2026.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025.

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4d video generation for robot manipulation. *arXiv preprint arXiv:2507.01099*, 2025.

Anson Lei, Frederik P. Nolte, Bernhard Schölkopf, and Ingmar Posner. Modular world models with competitive independent mechanisms. In *Causal and Object-Centric Representations for Robotics Workshop at CVPR*, 2024. URL https://openreview.net/forum?id=8exP12DDJo.

Atharva Sehgal, Arya Grayeli, Jennifer J. Sun, and Swarat Chaudhuri. Neurosymbolic grounding for compositional world models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.12690.

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. SPARTAN: A sparse transformer world model attending to what matters. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2411.06890.

Jack Brady, Roland S Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius Von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. In *International Conference on Machine Learning*, pages 3038–3062. PMLR, 2023.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable compositional generalization for object-centric learning. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=7VPTUWkiDQ.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation, 2023.

Jack Brady, Julius von Kügelgen, Sébastien Lachapelle, Simon Buchholz, Thomas Kipf, and Wieland Brendel. Interaction asymmetry: A general principle for learning composable abstractions. *arXiv preprint arXiv:2411.07784*, 2024.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. *Advances in Neural Information Processing Systems*, 36, 2024b.

Lingjing Kong, Shaoan Xie, Yang Jiao, Yetian Chen, Yanhui Guo, Simone Shao, Yan Gao, Guangyi Chen, and Kun Zhang. Learning by analogy: A causal framework for composition generalization. *arXiv preprint arXiv:2512.10669*, 2025.

Appendix

A Related Work

Structure in world models. A long line of work studies how to impose structure on representations and dynamics to improve generalization in world models and model-based control. We roughly group these efforts into four directions. First, state-abstraction methods learn compact representations that preserve control-relevant information, either by grouping states with similar rewards and transition behavior (Ferns et al., 2004; Castro, 2020; Zhang et al., 2021), or by learning invariant features useful for downstream control (Laskin et al., 2020). Second, model-based representation learning methods separate controllable or task-relevant factors from nuisance variation, so that latent dynamics focus on information that matters for prediction and planning (Fu et al., 2021; Wang et al., 2022a). Third, factored world-model methods make transition structure explicit by learning graph-structured dynamics over observed variables (Pitis et al., 2020; Wang et al., 2022b; Ji et al., 2024) or latent variables (Liu et al., 2023; Huang et al., 2022), following the factored MDP/POMDP perspective (Boutilier et al., 2000). Fourth, recent approaches incorporate geometric or physical structure into world models, including symmetry-preserving dynamics and geometry- or physics-consistent prediction (Lillemark et al., 2026; Zhang et al., 2026; Zhen et al., 2025; Liu et al., 2025).

Standard mixture-of-experts models route an input through selected parameter blocks, but routing alone does not expose which entities or directed dependencies support a transition. Recent structured world models address complementary parts of this problem. COMET learns reusable pairwise dynamics mechanisms through winner-takes-all competition and composes mechanism-context pairs in new environments (Lei et al., 2024). Cosmos binds object-centric neurosymbolic attributes to learned interaction rules for compositional generalization (Sehgal et al., 2024). SPARTAN sparsifies hard attention between object tokens to recover state-dependent local causal graphs and adapt to interventions (Lei et al., 2025). CWM instead targets massively multitask control: a language-conditioned mixture-of-experts provides coarse task-level sharing, while separate state-, action-, and task-conditioned entity and interaction masks select the slots and directed messages used for each target update. Thus, our contribution is not mixture-of-experts routing alone, but separating task-level expert selection from transition-level entity and message gating and evaluating their joint use across more than 200 tasks and held-out task recombinations.

Our work is related to these structured world-model approaches but addresses a different scaling regime. Rather than learning a compact abstraction for a single task or recovering a complete dynamics graph, we study massively multitask world models where each environment may reveal only part of a reusable concept library. The key challenge is not only to separate relevant from irrelevant factors, but also to reuse learned factors under held-out recombinations. To this end, CWM learns task-, entity-, and interaction-level routing masks that gate computational paths and pass the selected components through shared dynamics modules. This provides a lightweight structural bias for concept accumulation and recombination, without requiring full graph recovery or human-labeled latent factors.

Compositional Generalization Compositional generalization has been studied as the ability to recombine learned factors beyond the joint configurations observed during training. Existing theory often makes this problem tractable by imposing strong structure on how factors enter the observation model: Brady et al. (2023); Wiedemer et al. (2024a) analyze object-centric settings where concepts affect disjoint image regions, Lachapelle et al. (2023) assumes additive latent-to-observation effects, and Brady et al. (2024) extends this line by allowing limited interactions through interaction asymmetry. Wiedemer et al. (2024b) studies compositional generalization from first principles under access to the composition structure, while Kong et al. (2025) gives a causal view in which analogical transfer is enabled by shared relational mechanisms. Our empirical setting instead concerns large-scale multitask control, where a held-out task may recombine familiar objects, goals, action effects, and interaction patterns that were never jointly observed. We evaluate task-level routing and sparse entity- and interaction-level gates together with shared transition modules. Our theory has a narrower motivating role: it studies idealized source selection and illustrates mechanism reuse under explicit structural and coverage conditions, without claiming a generalization guarantee for the implemented learner.

B IDEALIZED SUPPORT SELECTION AND A RECOMBINATION ILLUSTRATION

B.1 PROOF OF THEOREM 3.2

Proof. The realizable oracle predictor has $\mathcal{L}_{\text{fit}} = 0$ and selects exactly the links in $\{\mathbf{M}_c^\star\}$. Now consider any zero-error admissible predictor. Fix a training regime c, target k, and required source $i \in \mathcal{I}_c^{(k)}$. If $M_{c,ki} = 0$, the hard-mask architecture makes its prediction a measurable function of $(\mathbf{z}_{-i,t}, \mathbf{a}_t, \mathbf{g}_e)$ within that regime: the local encoder f_i and every path from its input are excluded. Consequently, Assumption 3.1 implies

$$\mathbb{E}_{\mathcal{D}_c}\left[\|\hat{\mathbf{z}}_{k,t+1} - \mathbf{z}_{k,t+1}\|^2\right] \geq \Delta_{c,k,i} > 0, \tag{A1}$$

contradicting zero fit loss. Every exact-fit predictor must therefore contain every oracle link in every training regime.

The oracle attains support cardinality $\sum_c \|\mathbf{M}_c^\star\|_0$. Since every exact-fit predictor contains all these links, this is also the smallest feasible cardinality. Any additional link would increase that cardinality, so every solution of (6) has $\mathbf{M}_c = \mathbf{M}_c^\star$ for every training regime. □

Scope of the argument. The proof uses a positive population omission error, not a pair of exceptional points on the training support. It therefore requires only the almost-sure equality supplied by zero population loss. The partitions and regimes are given, the masks are fixed within each regime, and the oracle is feasible. No conclusion is drawn about arbitrary minima of a weighted fit-plus-sparsity loss, learned feature coordinates, or masks outside the observed regimes. In particular, predictive supports are not asserted to be causal graphs.

A concrete case satisfying predictive relevance. Within a fixed regime and at fixed action and context, consider a scalar target $Y = b + \sum_{i \in I} u_i(Z_i)$ with mutually independent source partitions. The local functions u_i may be nonlinear. If $0 < \mathrm{Var}(u_i(Z_i)) < \infty$ for every $i \in I$, then the optimal predictor omitting Z_i replaces its contribution by its mean, leaving $\Delta_i = \mathrm{Var}(u_i(Z_i)) > 0$. Thus independent source variation and a nonconstant local effect give a concrete sufficient case for Assumption 3.1. Independence and additivity are only features of this example, not requirements of the support proposition.

B.2 ILLUSTRATION: RECOMBINING COVERED BOUNDED-ORDER MECHANISMS

This separate sufficient case makes explicit one way to resolve the ambiguity in (8). It is an illustrative function class, not an additional architectural restriction imposed on CWM. Local mechanisms may be arbitrarily nonlinear, but their interaction order is bounded and their meanings are shared across regimes.

Shared mechanisms and their normalization. Let q collect the action and any context held fixed, excluding the coordinates being recombined. Fix an integer $r \geq 1$ and reference states $\mathbf{z}_i^0 \in \mathcal{Z}_i$. For each target k, consider

$$F_k^c(q, \mathbf{z}) = h_{k,\varnothing}(q) + \sum_{\substack{\varnothing \neq S \subseteq [K] \\ |S| \leq r}} A_{c,k,S}\, h_{k,S}(q, \mathbf{z}_S), \qquad A_{c,k,S} \in \{0,1\}. \tag{A2}$$

The functions $h_{k,S}$ are shared across regimes. The mechanism activations $A_{c,k,S}$ are supplied for this illustration. They select entire source subsets and are distinct from the source-target link masks $\mathbf{M}_c$. One special case is $A_{c,k,S} = \prod_{i \in S} M_{c,ki}$, which activates a mechanism when all its sources are retained.

For every nonempty S, both the true and fitted mechanisms obey the same anchoring restriction:

$$h_{k,S}(q, \mathbf{z}_S) = 0 \quad \text{whenever} \quad \mathbf{z}_i = \mathbf{z}_i^0 \text{ for some } i \in S. \tag{A3}$$

Thus a higher-order term cannot hide an effect of a proper subset of its sources. This shared normalization is a substantive restriction when activation patterns change, not merely a renaming of parameters.

Proposition B.1 (An illustrative covered-recombination result)**.** *Suppose the true and fitted predictors have the form (A2), with the same reference states, supplied activation patterns, and anchoring restriction (A3). Fix a test input $(q, \mathbf{z})$ and a supplied test activation pattern $A_\star$. For each target k, assume:*

1. *Training includes an all-reference configuration $(q, \mathbf{z}^0)$ for identifying the baseline.*
2. *For every S with $A_{\star,k,S} = 1$, a training regime c has $A_{c,k,S} = 1$ and includes every calibration configuration $(q, \mathbf{z}^B)$, $B \subseteq S$, where $\mathbf{z}_i^B = \mathbf{z}_i$ for $i \in B$ and $\mathbf{z}_i^B = \mathbf{z}_i^0$ otherwise. The regime activation pattern is unchanged across these configurations.*
3. *True and fitted responses agree at the all-reference configuration and every calibration configuration listed above.*

Then $\widehat{F}_k^\star(q, \mathbf{z}) = F_k^\star(q, \mathbf{z})$ for every target k.

Proof. At the all-reference configuration every nonempty mechanism vanishes, identifying $h_{k,\varnothing}(q)$. For a test-active subset S, choose a covering training regime c and take the alternating difference

$$\Delta_S F_k^c(q, \mathbf{z}) := \sum_{B \subseteq S} (-1)^{|S|-|B|} F_k^c(q, \mathbf{z}^B) = A_{c,k,S} h_{k,S}(q, \mathbf{z}_S). \tag{A4}$$

To verify the identity, anchoring removes a mechanism indexed by T unless $T \subseteq B$. For each $T \subseteq S$, its coefficient in the sum is $\sum_{B:\, T \subseteq B \subseteq S} (-1)^{|S|-|B|}$, which is zero for $T \subsetneq S$ and one for $T = S$. The baseline cancels as well. Since $A_{c,k,S} = 1$, agreement on the calibration configurations identifies $\widehat{h}_{k,S}(q, \mathbf{z}_S) = h_{k,S}(q, \mathbf{z}_S)$. Summing the identified baseline and mechanisms under $A_\star$ proves the claim. □

What coverage requires. For $r = 1$, the contrast subtracts the all-reference response from a single-source response. For $r = 2$, a pairwise mechanism requires four configurations, including joint variation of the two sources. Observing links involving $\{1, 3\}$ and $\{2, 3\}$ alone does not identify a new mechanism on $\{1, 2\}$. Every calibration point must use the test context q; the result does not cover arbitrary new actions or task embeddings. It applies to any fixed r, although when $r \geq K$ the coverage requirement can include the entire test configuration and offer no extrapolation benefit. Pointwise agreement in the proposition follows from zero population error if the true and fitted responses are continuous and the calibration points lie in the corresponding training supports; zero expected error alone is not sufficient at those points. The result identifies covered mechanism values, not the target activation pattern or the internal decomposition of a neural network.

An explicit complexity preference. For this illustrative class, a well-defined preference for fewer, lower-order mechanisms is

$$\Omega(h) = \sum_{k=1}^{K} \sum_{\substack{\varnothing \neq S \subseteq [K] \\ |S| \leq r}} \lambda_{|S|}\, \mathbf{1}\{h_{k,S} \not\equiv 0\}, \qquad 0 < \lambda_1 < \cdots < \lambda_r. \tag{A5}$$

The increasing weights penalize higher-order mechanisms more heavily. The covered-recombination proposition does not rely on minimizing this criterion: anchoring and calibration coverage identify the required values directly. Mechanisms never activated in training can be set to zero without altering training fit, but this does not justify their absence in an uncovered target. Neither this mechanism count nor its order bound is claimed to be enforced by CWM's soft-mask penalty.

Relation to the implemented model. Shared local and pairwise modules motivate this illustration, but pairwise message endpoints do not imply that the complete predictor has interaction order two. The task experts receive the full state, mask networks use global features, and the final update is nonlinear. These paths can introduce dependence on more than two partitions. The illustration therefore does not certify CWM's hypothesis class, recover its soft masks, or establish an approximate guarantee by merely calling higher-order effects a residual. Such an approximate guarantee would require explicit error control on the relevant calibration and target configurations.

(a) Held-out adaptation

Method	Score ↑
NewT	0.78
Dense-OC	0.69
Dense-OC + Task-MoE	0.73
CWM	**0.88**

(b) Unseen interaction unions

Mask	One-step MSE ↓
Oracle	0.12
Learned	0.19
Dense	0.45
Shuffled	0.39

Table A1: Controlled evaluation of routing and masking. (a) Adaptation to held-out MM-Bench tasks. (b) Prediction on unseen interaction unions in a system with known transition supports, evaluated without fine-tuning. Shuffled masks preserve row-wise sparsity while permuting source identities.

C Full Results

C.1 More Ablation Studies

Controlling for entity representations and expert capacity. We compare CWM with two controlled variants. Dense-OC uses the same entity representations, shared graph transition, training objective, planner, and interaction budget, but processes all entities and relations densely without task routing. Dense-OC + Task-MoE additionally uses the same task router and experts as CWM while retaining dense entity and interaction computation. Table A1(a) shows that CWM improves the held-out adaptation score from 0.73 to 0.88 over Dense-OC + Task-MoE, supporting the contribution of transition-dependent masking beyond entity representations and task-expert capacity.

Recombination with known interaction structure. We further evaluate a controlled system with six latent partitions and four reusable interaction modules. Training exposes each module individually; evaluation combines them into six unseen pairwise unions, without target-task training or fine-tuning. The learned soft masks achieve an AUPRC of 0.84 and an F1 score of 0.81. They also yield lower prediction error than dense masks and shuffled masks that preserve the number of selected sources per target (Table A1(b)). These results support predictive-support learning and recombination in this controlled setting.

C.2 Full Results on Multitask Training

Table A2 provides the full multitask training results, including both multi-task methods and single-task reference methods. Among multi-task agents, CWM achieves the best score in every domain and improves the aggregate average from 0.38 with NewT to 0.51. The gains are especially large in domains with object-centric or interaction-heavy structure, such as Meta-World, ManiSkill, RoboDesk, OGBench, and Atari, suggesting that task routing with sparse entity and interaction masking helps the world model focus on the factors that are predictive for each transition. Compared with multi-task TD-MPC2, the improvement is also substantial, indicating that simply scaling a strong model-based RL backbone across many tasks is not sufficient without additional structure for task routing and interaction masking.

The single-task columns are included only as references, since they train one separate agent per task and therefore do not face the same shared-representation challenge as multi-task methods. As expected, single-task TD-MPC2 remains strongest in most domains, but CWM narrows the gap while using a single shared agent across all tasks. This supports the main conclusion that compositional task routing and dynamics masking improve multi-task world-model learning without sacrificing the scalability of the NewT-style training pipeline.

C.3 Full Ablation Results

Figure A1 studies the contribution of the routing and masking components from §4 by removing each of them from the full model. We observe a consistent trend across all 10 domains: the full model performs best, and removing any one of these components leads to a noticeable degradation. Among the three components, task routing appears to be the most important. Removing it causes the largest drop in every domain, *e.g.*, from 0.69 to 0.44 on Meta-World, from 0.73 to 0.43 on ManiSkill, and from 0.51 to 0.34 on MiniArcade. This suggests that selecting task-relevant computational paths

Table A2: In-distribution final normalized scores at 100M environment steps on MM-Bench, averaged within each of the 10 domains. Values are mean ± std across seeds. "mt" denotes multi-task training, while "st" denotes single-task training. **Bold**: best multi-task method per row. Single-task methods are shown for reference.

Domain	BC st	PPO	FastTD3	TD-MPC2 mt	NewT	CWM	TD-MPC2 st
DMControl	$0.32_{\pm 0.02}$	$0.11_{\pm 0.05}$	$0.24_{\pm 0.04}$	$0.13_{\pm 0.03}$	$0.50_{\pm 0.04}$	$\mathbf{0.57}_{\pm 0.03}$	$0.81_{\pm 0.02}$
DMControl-Ext.	$0.24_{\pm 0.01}$	$0.17_{\pm 0.03}$	$0.13_{\pm 0.02}$	$0.13_{\pm 0.04}$	$0.37_{\pm 0.05}$	$\mathbf{0.47}_{\pm 0.04}$	$0.78_{\pm 0.03}$
Meta-World	$0.22_{\pm 0.03}$	$0.13_{\pm 0.05}$	$0.60_{\pm 0.06}$	$0.26_{\pm 0.04}$	$0.50_{\pm 0.03}$	$\mathbf{0.71}_{\pm 0.05}$	$0.79_{\pm 0.04}$
ManiSkill	$0.08_{\pm 0.02}$	$0.07_{\pm 0.04}$	$0.02_{\pm 0.01}$	$0.58_{\pm 0.05}$	$0.69_{\pm 0.04}$	$\mathbf{0.77}_{\pm 0.03}$	$0.92_{\pm 0.02}$
MuJoCo	$0.18_{\pm 0.04}$	$0.09_{\pm 0.03}$	$0.18_{\pm 0.05}$	$0.17_{\pm 0.06}$	$0.20_{\pm 0.03}$	$\mathbf{0.25}_{\pm 0.05}$	$0.82_{\pm 0.04}$
Box2D	$0.26_{\pm 0.03}$	$0.27_{\pm 0.06}$	$0.19_{\pm 0.05}$	$0.08_{\pm 0.02}$	$0.33_{\pm 0.03}$	$\mathbf{0.41}_{\pm 0.03}$	$0.86_{\pm 0.03}$
RoboDesk	$0.30_{\pm 0.02}$	$0.31_{\pm 0.05}$	$0.60_{\pm 0.03}$	$0.38_{\pm 0.03}$	$0.38_{\pm 0.04}$	$\mathbf{0.70}_{\pm 0.06}$	$0.82_{\pm 0.03}$
OGBench	$0.36_{\pm 0.04}$	$0.27_{\pm 0.05}$	$0.16_{\pm 0.03}$	$0.30_{\pm 0.04}$	$0.30_{\pm 0.03}$	$\mathbf{0.44}_{\pm 0.03}$	$0.93_{\pm 0.02}$
MiniArcade	$0.44_{\pm 0.03}$	$0.19_{\pm 0.06}$	$0.34_{\pm 0.07}$	$0.20_{\pm 0.04}$	$0.44_{\pm 0.03}$	$\mathbf{0.48}_{\pm 0.05}$	$0.94_{\pm 0.02}$
Atari	$0.18_{\pm 0.03}$	$0.07_{\pm 0.02}$	$0.09_{\pm 0.04}$	$0.08_{\pm 0.05}$	$0.13_{\pm 0.04}$	$\mathbf{0.28}_{\pm 0.05}$	$0.52_{\pm 0.06}$
Average	$0.26_{\pm 0.02}$	$0.17_{\pm 0.04}$	$0.26_{\pm 0.05}$	$0.23_{\pm 0.04}$	$0.38_{\pm 0.03}$	$\mathbf{0.51}_{\pm 0.04}$	$0.82_{\pm 0.03}$

Note: Highlighted yellow cells indicate the best result among multi-task methods. Pink columns indicate single-task methods, which use one agent per task and are shown only as references.

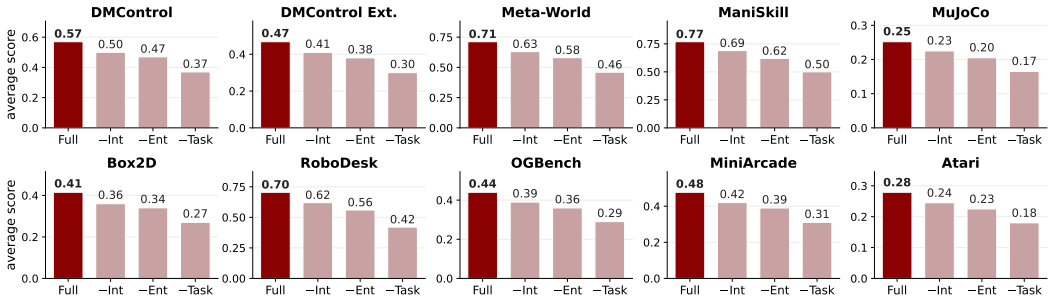

Figure A1: **Ablation of the proposed masks on different domains.** We compare CWM against three ablated variants that remove the interaction mask (-*Int*), entity mask (-*Ent*), or task routing (-*Task*), respectively.

is useful in large-scale multi-task training, where a dense shared representation can otherwise mix relevant and irrelevant information across tasks.

The entity mask also plays an important role. Its removal leads to substantial drops in domains with strong object-centric structure, such as Meta-World ($0.69 \rightarrow 0.56$), ManiSkill ($0.73 \rightarrow 0.58$), and MiniArcade ($0.51 \rightarrow 0.42$), supporting the usefulness of entity-level structure for multitask performance. The interaction mask provides a further, complementary improvement. Although its effect is somewhat smaller than that of task routing, removing it still consistently hurts performance, e.g., from 0.55 to 0.49 on DMControl, from 0.42 to 0.37 on DMControl-Ext., and from 0.38 to 0.33 on OGBench. This suggests that selective interaction routing is useful for learning across these domains.

Overall, these ablations show complementary contributions to multitask performance from task routing, entity masking, and interaction masking. They are consistent with the reuse hypothesis, but do not by themselves establish recovery of true mechanisms or behavior on unseen combinations. The routing and masking components serve complementary roles, and their combination yields the strongest and most stable performance across domains.

D Details of the NewT Backbone

Our method is built on a NewT multitask world model, which extends TD-MPC2 to language-conditioned and optionally image-conditioned continuous control. We summarize the backbone here for completeness. Given a language instruction $\mathbf{s}_{\text{lang}}$, optional RGB observation $\mathbf{s}_{\text{img}}$, and low-dimensional state observation $\mathbf{s}_{\text{state}}$, the model first computes

$$\mathbf{g} = \text{CLIP}_{\text{text}}(\mathbf{s}_{\text{lang}}), \qquad \mathbf{x} = \text{DINOv2}(\mathbf{s}_{\text{img}}), \qquad \mathbf{z} = h(\mathbf{s}_{\text{state}}, \mathbf{x}, \mathbf{g}), \tag{A6}$$

where $\mathbf{g}$ is the task embedding, $\mathbf{x}$ is the optional image embedding, and $\mathbf{z}$ is the latent state. In practice, the language and image encoders can be frozen pretrained backbones, while the remaining modules are learned MLPs.

The backbone contains four prediction heads:

$$\mathbf{z}' = d(\mathbf{z}, \mathbf{a}, \mathbf{g}), \qquad \hat{r} = R(\mathbf{z}, \mathbf{a}, \mathbf{g}), \qquad \hat{q} = Q(\mathbf{z}, \mathbf{a}, \mathbf{g}), \qquad \hat{\mathbf{a}} = p(\mathbf{z}, \mathbf{g}). \tag{A7}$$

Here, d is the latent dynamics model, R predicts rewards, Q predicts return targets, and p is a policy prior used to warm-start planning. Unlike generative world models that reconstruct future observations, this backbone uses a self-predictive latent objective, making the learned representation more directly tied to control.

World-model objective. Let $\tau = (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})_{t=0}^{H}$ be a replay subsequence sampled from buffer $\mathcal{B}$. The model predicts

$$\mathbf{z}'_{t+1} = d(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}), \qquad \hat{r}_t = R(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}), \qquad \hat{q}_t = Q(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g}), \tag{A8}$$

where $\mathbf{z}_t = h(\mathbf{s}_{\text{state},t}, \mathbf{x}_t, \mathbf{g})$. The world-model loss is

$$\mathcal{L}_{\text{wm}}(\theta) = \mathbb{E}_{\tau\sim\mathcal{B}}\left[\sum_{t=0}^{H} \lambda^t \left(\left\| \mathbf{z}'_{t+1} - \text{sg}(h(\mathbf{s}_{\text{state},t+1}, \mathbf{x}_{t+1}, \mathbf{g})) \right\|_2^2 + \ell_{\text{CE}}(\hat{r}_t, r_t) + \ell_{\text{CE}}(\hat{q}_t, q_t) \right)\right], \tag{A9}$$

where sg denotes stop-gradient, $\lambda \in (0, 1]$ downweights later rollout steps, and ℓ_{CE} is the discrete regression loss for rewards and values. Following TD-MPC2, the value target is computed by a one-step TD update,

$$q_t = r_t + \gamma Q_{\text{tgt}}\left(\mathbf{z}'_{t+1}, p(\mathbf{z}'_{t+1}, \mathbf{g}), \mathbf{g}\right), \tag{A10}$$

where Q_{tgt} is an exponential-moving-average target network. In practice, one can use an ensemble of Q networks and take the minimum over a random subset of target estimates, as in TD-MPC2.

Policy-prior objective. The policy prior p is trained with behavior cloning, value maximization, and entropy regularization:

$$\mathcal{L}_p(\theta) = \mathbb{E}_{\tau\sim\mathcal{B}}\left[\sum_{t=0}^{H} \lambda^t \left(\|p(\mathbf{z}_t, \mathbf{g}) - \mathbf{a}_t\|_2^2 - Q(\mathbf{z}_t, p(\mathbf{z}_t, \mathbf{g}), \mathbf{g}) - \mathcal{H}(p(\cdot|\mathbf{z}_t, \mathbf{g})) \right)\right]. \tag{A11}$$

The first term provides action supervision from demonstrations or planned actions, the second term encourages actions with high estimated return, and the third term maintains policy entropy. During online interaction, actions are selected by planning with the learned world model, with the planner warm-started by the policy prior.

Use of demonstrations. Following the NewT training protocol, demonstrations are used in several ways. First, before online interaction, the learnable model components are pretrained on demonstration trajectories using the world-model and policy-prior losses, with the value-maximization term in Equation (A11) optionally disabled during this stage. Second, planning can be initially biased toward the pretrained policy prior and then annealed toward unconstrained planning as online training progresses. Third, demonstration and online replay buffers can be sampled in balanced proportions so that demonstration data remain available throughout training. Finally, the behavior-cloning term in Equation (A11) continues to provide action supervision during online RL.

Connection to our method. Our method keeps the NewT-style training objective, reward/value heads, policy-prior loss, demonstration usage, and planning pipeline. The main change is in the latent dynamics model. Instead of using a single conditional transition $d(\mathbf{z}_t, \mathbf{a}_t, \mathbf{g})$, we introduce a task-routed, mask-conditioned transition that gates latent features and messages and composes them through shared dynamics modules. Thus, the backbone supplies a scalable multitask control objective, while our routing and masking components change how cross-environment structure is represented inside the latent dynamics.

E Implementation Details

Backbone and training protocol. We follow the NewT-style TD-MPC2 backbone for massively multitask world-model learning. Language instructions are encoded with a frozen CLIP-ViT/B text encoder, producing a 512-dimensional task embedding $\mathbf{g}$. When RGB observations are used, images are encoded with a frozen DINOv2/B encoder, producing a 768-dimensional visual feature. Low-dimensional state observations are embedded into a 128-dimensional state vector, and actions are padded or masked into a 16-dimensional action vector following the MMBench interface. Unless otherwise stated, the learnable backbone uses a 20M-parameter TD-MPC2-style world model with MLP components, LayerNorm, and Mish activations.

We keep the NewT training pipeline unchanged. The replay buffer capacity is 10^7 transitions, and replay subsequences are sampled uniformly. We use batch size 1024, update-to-data ratio 0.075, Adam optimizer, learning rate 3×10^{-4} for non-encoder modules, encoder learning rate 10^{-4}, and gradient clipping with norm 20. The temporal weighting coefficient is $\lambda = 0.5$. Reward and value predictions use discrete regression with 101 bins. The policy prior uses entropy coefficient 10^{-4}, and the behavior-cloning coefficient is 10. The target Q network is updated with EMA coefficient 0.99. For planning, we use horizon 3, 6 optimization iterations, population size 512, 64 elites, 24 policy-prior samples, minimum standard deviation 0.05, maximum standard deviation 2, and temperature 0.5. When constrained planning is used after demonstration pretraining, the constraint is annealed from 2M to 12M environment steps.

Task splits and evaluation. For in-distribution evaluation, all methods are trained on the standard MM-Bench multitask training set and evaluated on the same task families with held-out episodes. For compositional generalization, we construct held-out environments by recombining factors that are observed during training but not jointly observed in the same configuration. No trajectories from held-out compositional environments are used during source pretraining; target-task interaction data are used for the adaptation evaluation in Fig. 4. All reported scores are normalized following the MM-Bench protocol and averaged within each domain before computing the aggregate score.

Entity-level latent construction. For visual inputs, we instantiate entity-centric latent slots using a frozen segmentation-and-feature pipeline. We first apply SAM-3 to obtain object or region masks from the RGB observation. We keep the top N_{ent} masks according to mask confidence and area filtering; by default $N_{\text{ent}} = 6$. If fewer than N_{ent} valid masks are detected, we pad with dummy slots and mask them out in the transition module. If more masks are detected, we keep the highest-scoring non-overlapping masks.

Each masked region is encoded with DINOv2/B. We use the 768-dimensional DINOv2/B feature as the raw region descriptor, obtained by average pooling patch features inside the SAM mask. The region descriptor is concatenated with normalized mask geometry features, including bounding-box center, width, height, and mask area, and then projected to the entity-slot dimension:

$$\mathbf{z}_t^{(n)} = \mathrm{MLP}_{\text{ent}}\left([\mathrm{DINOv2}(\mathbf{o}_t \odot \mathrm{mask}_t^{(n)}),\ \mathrm{geom}(\mathrm{mask}_t^{(n)})]\right). \tag{A12}$$

We use a two-layer MLP with hidden dimension 512 and output dimension $d_{\text{ent}} = 128$. The resulting latent state is

$$\mathbf{z}_t = \{\mathbf{z}_t^{(n)}\}_{n=1}^{N_{\text{ent}}}. \tag{A13}$$

For state-only experiments, we use the same slot interface by projecting the compact NewT latent state into N_{ent} learned slots.

SAM proposals are not assumed to have a consistent ordering across frames or environments. For prediction targets, we treat the entity slots as an unordered set rather than as fixed-index latent variables. All entity-level modules share weights across slots, and summation is invariant to the order of its message inputs; the full transition need not be permutation-equivariant because task experts consume flattened slots. When visual slots are used as prediction targets, we compute the latent consistency loss after matching predicted and target slots with a permutation-invariant assignment loss. Thus, if an object appears as slot 1 in one observation and slot 2 in another, the model is not required to learn this arbitrary index correspondence; it only needs to predict the corresponding entity representation up to permutation.

Task-level mixture routing. The task router uses the CLIP task embedding $\mathbf{g}$ to select a sparse mixture of task experts. The router is a two-layer MLP with hidden dimension 256 and Mish

activation:

$$\boldsymbol{\alpha}^{\text{task}}(\mathbf{g}) = \text{TopK}\left(\text{softmax}\left(r_{\psi}^{\text{task}}(\mathbf{g})/\tau_{\text{moe}}\right)\right). \tag{A14}$$

By default, we use $J = 8$ task experts, activate the top $K_{\text{task}} = 2$ experts. The selected weights are renormalized to sum to one. Each expert is a two-layer MLP with hidden dimension 1024 and output dimension $d_{\text{task}} = 512$:

$$\mathbf{u}_{j,t}^{\text{task}} = E_j^{\text{task}}(\mathbf{h}_t, \mathbf{a}_t, \mathbf{g}), \qquad \mathbf{u}_t^{\text{task}} = \sum_{j=1}^{J} \alpha_j^{\text{task}}(\mathbf{g})\mathbf{u}_{j,t}^{\text{task}}, \tag{A15}$$

where $\mathbf{h}_t$ denotes the flattened structured latent representation. We use a load-balancing loss with coefficient $\beta_{\text{bal}} = 10^{-2}$ to prevent expert collapse. The task experts and router are trained end-to-end with the world-model objective; no task-expert labels are used.

Entity and interaction masking. Given entity slots $\{\mathbf{z}_t^{(n)}\}_{n=1}^{N_{\text{ent}}}$, action $\mathbf{a}_t$, and task-routed feature $\mathbf{u}_t^{\text{task}}$, we compute a local dynamics feature for each entity:

$$\mathbf{h}_t^{(n)} = f_{\theta}^{\text{dyn}}\left(\mathbf{z}_t^{(n)}, \mathbf{a}_t, \mathbf{u}_t^{\text{task}}\right). \tag{A16}$$

The local dynamics network f_{θ}^{dyn} is a two-layer MLP with hidden dimension 512 and output dimension 512. An entity mask network predicts a soft mask vector over entity slots:

$$\mathbf{m}_t^{\text{ent}} = \sigma\left(r_{\phi}^{\text{ent}}\left(\text{Pool}(\{\mathbf{z}_t^{(n)}\}_{n=1}^{N_{\text{ent}}}), \mathbf{a}_t, \mathbf{g}, \mathbf{u}_t^{\text{task}}\right)\right) \in [0,1]^{N_{\text{ent}}}, \tag{A17}$$

where r_{ϕ}^{ent} is a two-layer MLP with hidden dimension 512. Each scalar entry gates a vector entity feature:

$$\tilde{\mathbf{h}}_t^{(n)} = m_t^{\text{ent},(n)}\mathbf{h}_t^{(n)}. \tag{A18}$$

For pairwise interactions, we compute messages between entity pairs:

$$\mathbf{v}_t^{(n,\ell)} = \psi_{\theta}\left(\mathbf{z}_t^{(n)}, \mathbf{z}_t^{(\ell)}, \mathbf{a}_t, \mathbf{u}_t^{\text{task}}\right), \qquad n \neq \ell. \tag{A19}$$

The interaction network ψ_{θ} is a two-layer MLP with hidden dimension 512 and output dimension 512. The interaction mask network predicts a dense pairwise mask with scalar entries:

$$\mathbf{m}_t^{\text{int}} = \sigma\left(r_{\phi}^{\text{int}}\left(\text{Pool}(\{\mathbf{z}_t^{(n)}\}_{n=1}^{N_{\text{ent}}}), \mathbf{a}_t, \mathbf{g}, \mathbf{u}_t^{\text{task}}\right)\right) \in [0,1]^{N_{\text{ent}} \times N_{\text{ent}}}, \tag{A20}$$

where the diagonal entries are set to zero. The vector interaction message received by entity n is

$$\mathbf{c}_t^{(n)} = \sum_{\ell \neq n} m_t^{\text{int},(n,\ell)}\mathbf{v}_t^{(n,\ell)}. \tag{A21}$$

The next entity latent state is predicted by

$$\hat{\mathbf{z}}_{t+1}^{(n)} = u_{\theta}\left(\mathbf{z}_t^{(n)}, \tilde{\mathbf{h}}_t^{(n)}, \mathbf{c}_t^{(n)}, \mathbf{a}_t, \mathbf{u}_t^{\text{task}}\right), \tag{A22}$$

where u_{θ} is a two-layer MLP with hidden dimension 512 and output dimension d_{ent}. The full next latent state $\hat{\mathbf{z}}_{t+1}$ is obtained by concatenating the predicted entity slots and passing them to the reward, value, and policy heads through the standard NewT interface.

Routing and masking regularization. We train the task-routed, mask-conditioned transition with the NewT world-model loss and add sparsity and load-balancing regularization:

$$\mathcal{L}_{\text{wm}}^{\text{route}} = \mathcal{L}_{\text{wm}} + \mathbb{E}_{\tau \sim \mathcal{B}}\left[\sum_{t=0}^{H} \lambda^t \left(\beta_{\text{ent}}\|\mathbf{m}_t^{\text{ent}}\|_1 + \beta_{\text{int}}\|\mathbf{m}_t^{\text{int}}\|_1\right)\right] + \beta_{\text{bal}}\mathcal{L}_{\text{bal}}\left(\boldsymbol{\alpha}^{\text{task}}\right). \tag{A23}$$

Unless otherwise stated, we use $\beta_{\text{ent}} = 10^{-4}$, $\beta_{\text{int}} = 10^{-4}$, and $\beta_{\text{bal}} = 10^{-2}$. The ℓ_1 penalties encourage sparse entity and relation usage, while the load-balancing loss encourages the task router to use the expert set evenly across the multitask training distribution. All routing and masking components are trained end-to-end from latent prediction, reward prediction, value learning, and policy-prior losses.

Compute and implementation. We train all models for 100M environment steps unless otherwise specified. We use the same asynchronous environment interaction, replay-buffer sampling, demonstration pretraining, and model-based planning pipeline as the NewT backbone. For demonstration pretraining, we optimize the world-model and policy-prior objectives on the demonstration buffer before online interaction. During online training, demonstration and online replay buffers are sampled in balanced proportions, following the NewT protocol. For all experiments, we used either 8 L40 GPUs or 6 H100 GPUs.

F Broader Impact Discussion

This work is foundational research on compositional world models for reinforcement learning and is evaluated only in simulated control benchmarks. Its potential positive impact is to make model-based agents more sample-efficient and more systematic under new task compositions, which could reduce the amount of environment interaction, data collection, and compute required to adapt a controller to related tasks. By encouraging reusable dynamics structure rather than task-specific co-occurrence shortcuts, the method may also support more transparent analysis of which latent factors a model uses when predicting across environments.

At the same time, stronger world models and planners can create risks if they are incorporated into autonomous systems without domain-specific safety checks. For example, improved adaptation in robotics or other cyber-physical settings could amplify the consequences of modeling errors, unsafe objectives, distribution shift, or intentional misuse. The results in this paper should therefore not be interpreted as validating real-world deployment. Any downstream use should include task-specific hazard analysis, conservative deployment constraints, monitoring for distribution shift, human oversight where appropriate, and evaluation of privacy, fairness, and security concerns in the target application domain.

G License

Any derivative work built from existing task domains follows the license of the corresponding source domain. DMControl is licensed under the Apache 2.0 License, OGBench is licensed under the MIT License, and the Arcade Learning Environment (ALE; also known as Atari) is licensed under the GNU General Public License v2.0. MiniArcade and NewT are licensed under the MIT License.