

Belief Layers Over Temporal Knowledge Graphs Through the Lens of Model-Based Reinforcement Learning

Technical framing

The latent-state models in PlaNet and the Dreamer family separate **world state** from **observation** by maintaining a hidden model state and learning an observation-conditioned inference map into that state. In PlaNet, the Recurrent State Space Model (RSSM) combines a deterministic recurrent state h_t with a stochastic latent state s_t ; in DreamerV3, the world model remains an RSSM, but the stochastic part is sampled from a vector of categorical softmax distributions rather than a Gaussian. By contrast, TD-MPC2 uses a decoder-free, control-centric latent model: observations are encoded into a normalized latent representation, and the model predicts future latent quantities relevant for control without reconstructing future observations. ¹

That architectural difference matters for a belief layer above a temporal knowledge graph. Dreamer-style RSSMs expose an explicit distinction between a **predictive prior** and an **observation-conditioned posterior**, which is the closest analogue to a belief update in a temporal graph. TD-MPC2 exposes a distinction between **current encoded observation** and **imagined future latent rollouts**, but it does not maintain an explicit posterior distribution over hidden state in the same way. For a system whose primary requirement is belief-state semantics rather than decision-time control, Dreamer-style RSSM is therefore the more directly transferable template. This conclusion follows from the model interfaces described in the original papers. ¹

RSSM posterior between observations

In the original RSSM formulation, the latent dynamics are

$$h_t = f(h_{t-1}, s_{t-1}, a_{t-1}), \quad s_t \sim p(s_t | h_t), \quad o_t \sim p(o_t | h_t, s_t),$$

with an approximate filtering posterior

$$q(s_t | h_t, o_t).$$

PlaNet states this explicitly and parameterizes the approximate posterior as a **diagonal Gaussian**. The model uses the filtering posterior for planning, while noting that a full smoothing posterior can also be used during training. ²

DreamerV3 preserves the same prior/posterior split but changes the form of the stochastic state. Its encoder produces stochastic representations sampled from a **vector of softmax distributions**, with straight-through gradients through sampling; the model state is the concatenation of the recurrent state

and the stochastic representation. The dynamics loss is the KL divergence between the dynamics predictor and the next stochastic representation, and the representation loss regularizes the stochastic representation to remain predictable. DreamerV3 also mixes each categorical distribution with 1% uniform mass to prevent the encoder and dynamics predictor from becoming deterministic. ³

Between observation points, the RSSM does not maintain a new observation-conditioned posterior unless a new observation arrives. It propagates the **predictive prior** forward through the transition model. In PlaNet, this distinction is visible in the one-step KL term $\text{KL}[q(s_t | o_{\leq t}, a_{< t}) || p(s_t | s_{t-1}, a_{t-1})]$, and PlaNet’s latent-overshooting objective was introduced precisely because planning depends on accurate multi-step priors when intermediate observations are unavailable. ⁴

TD-MPC2 is different in a way that is directly relevant to a belief store. The paper describes its world model as **implicit** and **decoder-free**: observations are encoded into a normalized latent representation, and the model recurrently predicts actions, rewards, and values without decoding future observations. The paper does not define a separate Bayesian posterior $q(z_t | o_t, \dots)$ and prior $p(z_t | z_{t-1}, a_{t-1})$ interface of the Dreamer/PlaNet kind. It therefore provides less direct support for representing calibrated belief uncertainty over hidden state. ⁵

What RSSM uncertainty means

Vanilla RSSM does **not** introduce distinct symbolic states for “stale,” “contradicted,” and “under-observed.” Instead, those cases are mediated through the same latent-state machinery: a predictive prior, an observation-conditioned posterior, and a KL divergence that encourages them to align. A contradiction appears as a large mismatch between prior prediction and posterior update; staleness appears as uncertainty that accumulates while the model rolls the prior forward without new evidence; under-observation appears as a posterior that remains broad because the current observation does not identify the latent state sharply. This is an inference from the RSSM definitions and losses in PlaNet and DreamerV3, not a separate mechanism named in those papers. ⁶

The uncertainty literature around RSSM makes this limitation explicit. Becker and Neumann argue that RSSMs overestimate **aleatoric** uncertainty and rely on that overestimation as an implicit regularizer, rather than cleanly separating aleatoric and epistemic uncertainty. They further note that this becomes problematic when uncertainty calibration matters, including occlusions, missing observations, and sensor modalities with different update frequencies. ⁷

When model-based RL papers want an explicit decomposition, they usually add extra structure outside vanilla RSSM. PETS uses **probabilistic ensembles with trajectory sampling** to obtain uncertainty-aware dynamics and uncertainty propagation. Becker and Neumann’s VRKN combines Kalman-style latent updates for smoothing with Monte Carlo dropout for epistemic uncertainty, and they report that this design handles missing observations and varying observation frequencies more naturally than RSSM. ⁸

TD-MPC2 also does not solve the stale-versus-contradicted-versus-under-observed distinction at the latent-state level. Its ensemble mechanism is on the **Q-functions** used for value estimation, while the world model itself remains an implicit latent predictor rather than an explicit posterior belief model. For a semantic belief layer, that means the distinction must be introduced by the surrounding data model rather than expected to emerge from vanilla TD-MPC2 alone. ⁹

Transfer to irregular event-sourced knowledge graphs

A direct transfer from continuous-observation MBRL is the **prior/update separation** itself: maintain a predictive belief state between events, and apply an update only when a new event arrives. However, vanilla RSSM assumes discrete-time sequences $\{o_t, a_t, r_t\}$ indexed by uniform steps. For irregular event streams, elapsed real time must be represented explicitly. Continuous-time latent models such as Latent ODEs and Neural CDEs were introduced specifically because standard RNN-style models do not handle arbitrary time gaps well; Latent ODEs and ODE-RNNs can handle arbitrary gaps and can also model observation-time probabilities via Poisson processes. GRU-D shows a second transferable pattern: pass both a **mask** and **time since last observation** into the recurrent update, because missingness itself can be informative. ¹⁰

For an event-sourced temporal graph, the observation model should not reconstruct pixels. It should emit likelihoods over **typed semantic facts**. DreamerV3 already treats the world model as a collection of prediction heads over rewards, continuation flags, and reconstructed inputs; TD-MPC2 goes further and removes observation reconstruction entirely, predicting only control-relevant latent outcomes. A direct adaptation is therefore to replace image decoders with predicate-specific likelihood heads, such as Bernoulli or categorical heads for individual facts, count heads for multiplicities, and relation-specific heads for entity pairs. This adaptation is a design extrapolation from the modular head structure in DreamerV3 and TD-MPC2. ¹¹

For state transitions that were never directly observed, the transferable idea is **smoothing** rather than filtering alone. PlaNet explicitly notes the distinction between filtering and smoothing, and VRKN uses exact smoothing inference in latent space. In a temporal knowledge graph, that corresponds to representing a hidden transition as an **inferred interval event** or a posterior over transition time bounded by surrounding observations, rather than writing it into the graph as if it had been directly observed. ¹²

A second transfer is to model **event emission** separately from **state persistence**. Latent ODEs explicitly model observation-time probabilities, and neural point-process models represent irregular event dynamics using latent processes that govern event intensity and event-sequence uncertainty. For a belief store, that suggests a predicate-specific hazard or intensity model for “how surprising is it that no new event has arrived yet?” That is the mechanism needed to distinguish an aging-but-credible belief from one that should already have emitted a correcting event by now. ¹³

A third transfer is to optimize for **multi-step latent consistency** rather than only one-step reconstruction. PlaNet’s latent overshooting was designed to train multi-step priors directly, and TD-MPC2’s joint-embedding prediction similarly trains latent rollouts beyond raw reconstruction. In sparse event regimes, that is directly applicable because long periods may pass without any event carrying corrective evidence. ¹⁴

Multiple objects and combinatorial growth

The published DreamerV3 RSSM is holistic: it uses a recurrent state and a stochastic representation, but not explicit object slots. Recent object-centric MBRL papers augment or replace that holistic latent with object-level structure. Slot Attention produces **exchangeable slots** that bind to objects through competitive attention; SOLD builds a model-based RL pipeline on top of object slots and uses transformer-based

dynamics over slot sets; OC-STORM injects object representations from segmentation into a world model so that model capacity is focused on decision-relevant entities and interactions. ¹⁵

The core technique for reducing combinatorial growth is **factorization by objects plus sparse interaction modeling**. G-SWM decomposes a scene into a set of object latents and a context latent, gives each object state and attribute components, and propagates only the top- K objects forward. Object dynamics are modeled explicitly with an object-state RNN, while rendering uses object attributes plus a separate context channel. This avoids enumerating all joint scene configurations as a single monolithic state, and instead models a bounded set of interacting object states plus background context. ¹⁶

Structured World Belief extends this in a way that maps directly onto the knowledge-graph belief problem. SWB argues that existing object-centric world models lack **object permanence** and **belief states**. It separates an object's **presence in belief** from its **visibility in observation**, preserves files for invisible objects, uses file-slot matching to reattach observations to persistent latent objects after long occlusions, and maintains multiple weighted particle hypotheses through sequential Monte Carlo. That is precisely the pattern required when multiple entities evolve concurrently and some of their state transitions are not directly observed. ¹⁷

Object-centric factorization does not eliminate combinatorial growth entirely. It replaces symbolic cross-products with approximate factorization under a **fixed slot or object-file budget**, plus attention- or message-passing-based interaction models. SOLD's transformer dynamics over object slots and G-SWM's top- K propagation make this explicit. The result is reduced state entanglement and improved compositionality, but not exact scaling with arbitrary numbers of real-world entities. This scaling characterization is an inference from the architectures in those papers. ¹⁸

Concrete design for a belief store

A belief store for a temporal knowledge graph should not collapse everything into a single confidence number. The literature supports separating at least four axes: **truth status**, **observation status**, **temporal freshness**, and **uncertainty decomposition**. The open-world assumption in knowledge bases states that absent statements are **unknown**, not false, while truth-maintenance systems are explicitly designed to store assumptions and revise them when discoveries contradict them. SWB adds a further separation that is directly transferable: an entity can remain present in belief while being absent from current observation. ¹⁹

A concrete schema is: 1. **truth_status**: supported, contradicted, unknown, inconsistent; 2. **origin_status**: directly_observed, smoothed_inference, predictive_persistence, never_observed; 3. **coverage_status**: visible_or_reported, not_reported, outside_local_completeness_scope; 4. **uncertainty fields**: one for predictive spread or entropy, one for evidence sparsity, and optionally one for model disagreement if ensembles are used. This decomposition is the direct way to represent “not yet observed” separately from “observed as contradicted,” which vanilla RSSM does not do internally. ²⁰

Each belief update should store both a **prior** and a **posterior**. Operationally, this means preserving the predicted state before conditioning on a new event and the updated state after conditioning on that event, together with a divergence or surprise score. Large prior/posterior mismatch is the correct signal for

contradiction or revision; monotonically increasing prior uncertainty without counterevidence is the correct signal for staleness; high posterior uncertainty with little evidence is the correct signal for under-observation. This mirrors the prior/posterior split in RSSM and makes the three cases queryable instead of latent. ⁶

Transitions that were never directly observed should be represented as **latent transition beliefs**, not as synthetic direct observations. If fact A is supported at t_1 and incompatible fact B is supported at t_2 , the correct stored object is a transition hypothesis with support interval $(t_1, t_2]$, posterior mass over possible change times, and provenance attached to both boundary observations. Offline smoothing can then revise that transition-time posterior as later facts arrive. This design follows directly from state-space smoothing rather than filtering-only inference. ¹²

Aging should be predicate-specific. Some beliefs are **persistence-dominant** and remain stable until contradicted; others are **emission-dominant** and should receive frequent confirming or changing events. The belief store should therefore maintain a per-predicate persistence or hazard model, using elapsed time since the last supporting event as an input. Continuous-time latent models and point-process models provide the template: state can evolve continuously, while event arrival intensity is separately modeled. That lets the system represent “credible but aging” as distinct from “likely stale because this predicate usually emits again by now.” ¹³

“Observed as contradicted” should require **explicit negative support** or a **local completeness assertion**, not mere absence. Open-world KB semantics state that absence means unknown. The KG literature lists the mechanisms that legitimately create negative knowledge: explicit negative predicates, no-value objects, deprecated or false statements, and completeness/cardinality information that closes a local subject-predicate set. In a belief store, contradiction should therefore be entered only when there is direct counterevidence or a completed local frame that licenses negation. ²¹

For concurrency and scale, factor the belief layer by **entity-local state chains** plus **relation factors**, rather than by whole-graph snapshots. Use persistent entity files or slots, store existence separately from current observability, and retain multiple hypotheses when entity identity or transition timing is ambiguous. This directly follows the object-permanence and particle-hypothesis design in SWB and the slot-based factorization in object-centric world models. ²²

The resulting architecture is structurally close to an RSSM-based filter/smoother, but with semantic metadata that vanilla RL world models do not provide. The pattern is: immutable event log below; predictive prior above; posterior update on each new event; optional backward smoothing for hidden transitions; explicit negative-evidence and completeness channels for contradiction; and per-entity object permanence for long gaps and concurrent trajectories. That combination matches the belief requirements of an event-sourced temporal knowledge graph more closely than using DreamerV3 or TD-MPC2 without modification. ²³

¹ ² ⁴ ⁶ ¹⁰ ¹² ¹⁴ ²⁰ ²³ <https://proceedings.mlr.press/v97/hafner19a/hafner19a.pdf>
<https://proceedings.mlr.press/v97/hafner19a/hafner19a.pdf>

³ ¹¹ <https://arxiv.org/html/2301.04104>
<https://arxiv.org/html/2301.04104>

5 9 <https://arxiv.org/html/2310.16828v2>

<https://arxiv.org/html/2310.16828v2>

7 <https://arxiv.org/abs/2210.09256>

<https://arxiv.org/abs/2210.09256>

8 <https://arxiv.org/abs/1805.12114>

<https://arxiv.org/abs/1805.12114>

13 <https://arxiv.org/abs/1907.03907>

<https://arxiv.org/abs/1907.03907>

15 <https://arxiv.org/abs/2006.15055>

<https://arxiv.org/abs/2006.15055>

16 <https://proceedings.mlr.press/v119/lin20f/lin20f.pdf>

<https://proceedings.mlr.press/v119/lin20f/lin20f.pdf>

17 22 <https://ar5iv.labs.arxiv.org/html/2107.08577>

<https://ar5iv.labs.arxiv.org/html/2107.08577>

18 <https://arxiv.org/html/2410.08822v1>

<https://arxiv.org/html/2410.08822v1>

19 21 <https://suchanek.name/work/publications/acm-csur-2023.pdf>

<https://suchanek.name/work/publications/acm-csur-2023.pdf>