

Temporal Knowledge Graph Design Lessons for a Belief-Enhanced Event-Sourced Fact Graph

Executive synthesis

The temporal KG literature separates three concerns that your architecture already hints at but should keep distinct: **event capture**, **state materialization**, and **belief over state**. In the benchmark and modeling literature, ICEWS and GDELT are treated as **episodic/event** temporal KGs with timestamped interactions, while YAGO-style datasets are treated as **valid-time fact** KGs with intervals or interval boundaries. Embedding models such as TComplEx, TNTComplEx, TeRo, TimePlex, and Temporal TransE mostly learn **time-conditioned plausibility scores** for quadruples; they do **not** provide a native semantics for contradiction management, explicit revision history, or the distinction between “still true but stale” and “explicitly terminated.” The temporal database literature adds the missing semantics: valid time, transaction time, reference time, open-ended intervals, and time-slice queries. For a belief-enhanced event-sourced system, the strongest fit is therefore an **append-only event spine with explicit begin/end or supersession events**, a **materialized anchor layer** for current and as-of state, and a **belief layer** that carries confidence, decay, calibration, and revision separately from truth maintenance. ¹

In that framing, the main result from the literature is not that one embedding family is “the right model” for your system. The result is narrower: **interval-bounded or event-bounded truth semantics** align with an append-only spine, while **decay** aligns with the belief layer. Learned decay is useful for ranking or freshness priors, but it is weak as a truth semantics because it collapses “unrefreshed” and “refuted” unless paired with an explicit invalidation or conflict model. ²

Temporal validity encodings in the main embedding models

Across the five model families you named, the dominant pattern is **timestamp modulation**, not event-sourced invalidation. TComplEx extends ComplEx by adding a timestamp embedding to an order-4 tensor decomposition, so a fact is scored as a function of subject, predicate, object, and time. TNTComplEx adds a **non-temporal component** on top of the temporal one so predicates can partly ignore time when appropriate. This is useful when a corpus mixes genuinely temporal relations with mostly static ones, but the truth semantics remain “score this quadruple at time t ,” not “this event starts or ends a state.” ³

Temporal TransE, in the “Deriving Validity Time in Knowledge Graph” line of work, also remains point-oriented. Its variants encode time either as a **synthetic time-specific relation**, as a **vector in the same space as entities and relations**, or as a **scalar coefficient** that modulates embeddings. In all cases, time is attached to a scored quadruple; there is no native notion of an append-only begin/end event pair, no explicit invalidation event, and no learned staleness semantics beyond the score surface itself. That makes Temporal TransE compatible with an append-only spine only after a preprocessing step that flattens events or interval boundaries into timestamped quadruples. ⁴

TimePlex is more interval-aware than TComplex or Temporal TransE, but only indirectly. It treats temporal KB facts as holding over **time intervals** and scores an interval by summing per-time-instant scores; it then augments the base embedding with learned **recurrence**, **ordering**, and **time-gap** constraints, using Gaussian distributions over recurrence gaps and relation-pair gaps. This is closer to an append-only event spine because recurring and ordered events can be learned from historical change points, but its inference still operates as timestamp ranking plus interval coalescing rather than as explicit event-based invalidation. The model's semantics are therefore “the fact is plausible over these instants,” not “a later event formally superseded the earlier state.” 5

TeRo is the closest of the listed models to explicit state boundaries. It defines temporal evolution as a **rotation** of entity embeddings in complex space and, for facts involving intervals, represents each relation with a **pair of dual complex embeddings** to handle the **beginning** and **end** of the relation separately. The paper also argues this is more efficient than discretizing a span into all intermediate timestamps, because a fact with interval $[t_b, t_e]$ can be represented via boundary quadruples rather than all internal time points. For an append-only event spine, this boundary-based view is a better match than per-instant scoring because begin and end can map naturally to monotonic state-change events. 6

The compatibility ranking for an append-only event spine is therefore an inference from the papers, not a claim they make directly. On that inference: **TeRo** is the best fit among the listed models if you want truth to be induced from explicit boundaries; **TimePlex** is next if you want interval prediction with learned temporal constraints; **TComplex/TNTComplex** fit if you materialize timestamped anchors or snapshots; **Temporal TransE** fits only after stronger flattening and loses the distinction between “start,” “end,” and “unchanged state.” None of the five natively uses **event-based invalidation** as the primary truth representation, and none of them natively models **belief staleness** as distinct from contradiction. 7

Semantics of current under mixed granularity

The temporal database literature is clear that “**current**” is **underspecified unless the time axis is named**. The important distinction is between **valid time** (when a fact is true in the modeled world), **transaction time** (when the system learned or stored it), and **reference time** (the observer’s frame of reference for “now” or for time travel). The classic result is that a database with “now” variables needs an explicit semantics because intuitive use of “current” otherwise leads to ambiguity and errors. For your system, that means “current anchor” should not mean only “latest sequence number.” It should mean at least: **valid at reference time rt^*** , given all events ingested through **transaction time tt^*** . 8

Mixed granularity makes this stricter, not looser. The multi-granularity TKG literature shows that single-granularity temporal representations mislead reasoning when questions and facts mix year, month, and day-level constraints, and that adaptive granularity handling improves performance. Independent work on temporal data models with heterogeneous granularities argues that valid times can be expressed at different levels of abstraction and often carry **indeterminacy**; one response is to use explicit interval structure and **three-valued logic** so the system can represent “true,” “false,” and “undetermined at this granularity” rather than forcing false precision. 9

For your architecture, the practical consequence is that “current” should be **relation- or anchor-local**, not global. An anchor refreshed every task run and an anchor that legitimately persists for months should both be current if their last state remains valid at the chosen reference time and there is no terminating event. Their difference belongs in metadata such as native granularity, expected refresh cadence, or learned

volatility, not in the truth predicate itself. The anchor state should therefore be represented as an **open-ended valid-time interval** at its native granularity, optionally coalesced across adjacent equal states, while the belief layer may assign lower freshness to fast-changing anchors than to slow-changing ones. That recommendation is an inference from the literature on “now,” bitemporality, coalescing, and mixed granularity. ¹⁰

A compact semantics that matches the literature is: **current(anchor, rt^* , tt^*) := the maximal coalesced anchor state whose valid interval contains rt^* , computed from all events with transaction time $\leq tt^*$.** This gives you a principled way to handle late-arriving corrections, historical replay, and “as believed then” versus “as now known.” ⁸

Persistent facts, transient events, and the anchor-versus-spine distinction

The event-based TKG literature often draws the same distinction your architecture makes, but under different names. One line explicitly contrasts **episodic** temporal KGs with **semantic** KGs and tests whether current semantic facts can be derived from episodic memory by projection. In that work, ICEWS and GDELT are modeled as episodic event streams, and current semantic facts are described as something that must be derived from those timestamped events, not stored as the same object. The paper further notes that deriving current semantics from ICEWS requires considering both **starting** and **terminal** points of consecutive events. ¹¹

ICEWS and GDELT are therefore closest to your **spine**. ICEWS is described as timestamped political or socio-political interactions between actors, and standard benchmark derivatives treat it as event facts with single timestamps or daily granularity. GDELT’s Event Database similarly records more than 300 categories of physical activities around the world, while its separate Global Knowledge Graph tracks contextual entities, themes, and emotions from news reports. In both cases, the primary object is an **event/report at a time**, not a first-class persistent state assertion with explicit begin and end validity. ¹²

YAGO15K is closer to your **anchor layer**. The temporal KG completion literature describes YAGO-style data as facts that may hold for a **time interval** or a **specific point in time**, and the YAGO15K benchmark is repeatedly described as containing temporal modifiers such as **occursSince** and **occursUntil**. That makes it fundamentally a **valid-time fact benchmark** rather than an event log. Unlike ICEWS and GDELT, it natively represents persistent relations with duration, even if many models preprocess those intervals into start/end points or sampled timestamps. ¹³

The mapping to your terminology is therefore direct. **Spine facts** correspond to episodic/event quadruples as in ICEWS and GDELT. **Anchor facts** correspond to projected, interval-scoped, or current-valid facts as in YAGO-style data or the semantic projection derived from episodic data. The literature supports keeping these layers separate: event logs for causal/history preservation, and state/semantic layers for retrieval and reasoning about what currently holds. ¹⁴

Query patterns emphasized by the literature that are not explicit in your description

The literature supports several temporal query classes that are not explicit in the system you described. The first is the **time-slice or snapshot query**: “what was true at time T ?” The temporal database literature formalizes this via valid-time and transaction-time slices, and the “now” literature explicitly introduces reference time to support “time travel” views of the database. Your current description mentions lineage from the current anchor back to originating events, but not arbitrary materialization of the graph at any valid/reference time pair. ⁸

The second is **time prediction and interval prediction**. TimePlex evaluates $(s, r, o, ?)$ interval prediction and proposes time-aware evaluation for partially overlapping intervals. TEILP goes further by converting a TKG into a temporal event KG and attaching **conditional probability density functions** to learned rules for interval/time prediction. This is a different query pattern from anchor lineage: it asks not only what caused the current state, but also **when** a fact likely held or will hold. ¹⁵

The third is **multi-hop temporal logic queries**. TFLEX defines complex reasoning over TKGs where the answer can be an entity or a timestamp and adds temporal operators such as **After**, **Before**, and **Between** on top of first-order logical operations. The TKGQA survey and MultiTQ benchmark also highlight question types such as **Before/After**, **First/Last**, **Timejoin**, and mixed-granularity temporal constraints. If your system currently supports only current-anchor lookup plus backward lineage traversal, these are materially richer query classes. ¹⁶

The fourth is the **temporal join**. Temporal database work treats joins over valid-time intervals as a first-class operator and distinguishes ordinary temporal joins from **durable temporal joins**, which require the overlap interval to be long enough that the result is not just a transient coincidence. In a fact graph context, this supports queries such as “which entities simultaneously satisfied relation A, relation B, and relation C over a nontrivial interval?” That capability is not implied by current-anchor materialization alone. ¹⁷

The fifth is **forecasting** rather than interpolation. TLogic learns explainable temporal logical rules for **future** link forecasting, not only reconstruction of known historical facts. Know-Evolve and Graph Hawkes Neural Network model event occurrence in continuous time using point-process machinery, allowing next-event or recurrence-time prediction. Those are useful if your belief layer will eventually need to represent expected future state changes or reminder-style priors, but they are not substitutes for truth maintenance in the current anchor graph. ¹⁸

Belief staleness, contradiction, and the right division of labor

The literature supports three different mechanisms that are often conflated: **validity intervals**, **event-based invalidation**, and **temporal decay**. They do different jobs. Validity intervals say **when a fact is true**. Event-based invalidation says **which event caused truth to stop holding or be superseded**. Decay says **how much confidence or relevance should be assigned as time passes**. For a belief system that must distinguish **stale** from **contradicted**, these mechanisms should not be merged. ¹⁹

If the goal is to distinguish stale beliefs from contradicted ones, **event-based invalidation plus valid-time intervals** gives the cleanest semantics. A belief is **contradicted** when a later event or conflict rule closes its

valid interval, supersedes it, or shows it cannot coexist with another fact under temporal constraints. Work on uncertain temporal KGs and temporal conflict resolution explicitly treats temporal conflicts and conflict-free temporal KGs as a separate reasoning problem. That is close to your “revision” concept, and it belongs above or alongside the event spine, not inside a decay function. ²⁰

By contrast, **decay-based representations** are suitable for stale beliefs but weak for contradiction. HALO quantifies temporal validity via a half-life-based decay function to filter outdated facts, and TempValid models the temporal validity of rule confidence as a learned time function. These are useful when the system should downweight old evidence or old rules, but they do not by themselves say whether a fact was **refuted** or merely **not refreshed**. If decay is stored in the belief layer and the underlying truth state remains open-ended until explicitly terminated, the stale-versus-contradicted distinction stays intact. If decay is used as the truth criterion, the distinction becomes ambiguous. ²¹

Point-process and recurrence models occupy a third niche. Know-Evolve models event occurrence as a multivariate point process whose intensity depends on learned embeddings, and GHNN extends Hawkes-style reasoning to evolving graph sequences. These models are useful for forecasting **when another event is likely to happen**, especially recurring events, but the semantics is event hazard or recurrence intensity, not truth status. They are therefore appropriate as priors for “this belief is getting stale because a refresh event was expected by now,” but not as the primary mechanism for contradiction. ²²

The resulting design recommendation is straightforward. Put **truth maintenance** in the append-only event spine plus materialized valid-time anchors. Put **belief maintenance** in a separate layer with confidence, calibration, half-life/freshness, and revision history. In that design, a belief can move from high confidence to low confidence without any contradiction event, and it can be cleanly contradicted when an explicit invalidating event or conflict rule arrives. That division of labor is the most faithful synthesis of the temporal KG, forecasting, and temporal database literature. ²³

Open questions and limitations

The literature is still weak on a few points that matter directly to your architecture. First, most TKG embedding work optimizes **link completion or forecasting**, not explicit **belief revision semantics** with auditable contradiction handling. Second, benchmark families are mismatched: ICEWS and GDELT emphasize event streams, while YAGO-style benchmarks emphasize interval-scoped facts, and neither exactly matches a monotonic global-sequence event spine with anchor lineage and a calibrated belief layer. Third, mixed granularity is now recognized as important, but relation-specific volatility and uncertainty semantics remain comparatively underdeveloped in mainstream TKG benchmarks and models. ²⁴

The highest-confidence conclusion is therefore architectural rather than model-specific: use the TKG literature for **temporal scoring, interval inference, and richer query patterns**, but keep **event-sourced truth, state materialization**, and **belief decay/calibration** as separate layers with separate semantics. That is the combination most consistent with the surveyed work. ²⁵

- 2 21 HALO: Half Life-Based Outdated Fact Filtering in Temporal Knowledge Graphs
https://arxiv.org/abs/2505.07509?utm_source=chatgpt.com
- 3 7 25 openreview.net
<https://openreview.net/pdf?id=rke2P1BFwS>
- 4 (PDF) Deriving Validity Time in Knowledge Graph
https://www.researchgate.net/publication/324629547_Deriving_VValidity_Time_in_Knowledge_Graph
- 5 15 Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols
<https://aclanthology.org/2020.emnlp-main.305.pdf>
- 6 [2010.01029] TeRo: A Time-aware Knowledge Graph Embedding via Temporal Rotation
<https://arxiv.org/abs/2010.01029>
- 8 10 www2.cs.arizona.edu
<https://www2.cs.arizona.edu/~rts/pubs/TODS97.pdf>
- 9 [2408.15293] Learning Granularity Representation for Temporal Knowledge Graph Completion
<https://arxiv.org/abs/2408.15293>
- 12 [1809.03202] Learning Sequence Encoders for Temporal Knowledge Graph Completion
<https://arxiv.org/html/1809.03202>
- 13 Learning Sequence Encoders for Temporal Knowledge Graph Completion - ACL Anthology
<https://aclanthology.org/D18-1516/>
- 16 TFLEX: Temporal Feature-Logic Embedding Framework for Complex Reasoning over Temporal Knowledge Graph
https://arxiv.org/abs/2205.14307?utm_source=chatgpt.com
- 17 Computing Complex Temporal Join Queries Efficiently
<https://cs.uwaterloo.ca/~xiaohu/papers/sigmod22-temporal.pdf>
- 18 TLogic: Temporal Logical Rules for Explainable Link Forecasting on Temporal Knowledge Graphs | Proceedings of the AAAI Conference on Artificial Intelligence
<https://ojs.aaai.org/index.php/AAAI/article/view/20330>
- 19 20 23 (PDF) Towards Probabilistic Bitemporal Knowledge Graphs
https://www.researchgate.net/publication/324634120_Towards_Probabilistic_Bitemporal_Knowledge_Graphs
- 22 Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs
https://arxiv.org/abs/1705.05742?utm_source=chatgpt.com