THE UNIVERSITY *of* EDINBURGH

From Post-Hoc Fixes to Built-In Refinement in Attention Steering and Agentic Memory

Waylon Li

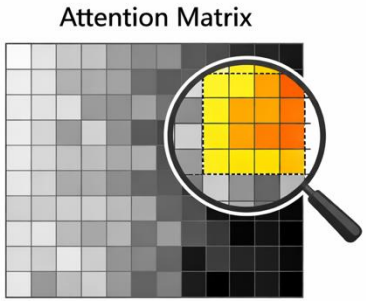
AIAI, School of Informatics
University of Edinburgh

29th May, 2026

Two representational learning work in two distinct tasks

The **cat** sat on the mat and then chased after a small mouse.

Focusing on the phrase: "chased after a small mouse".



High attention weights

Attention Steering

Given user-marked spans of the prompt, control the share of attention these tokens received so that the model respects the user's emphasis without retraining

Agentic Memory for temporal reasoning

Given a growing memory of facts that may become invalid over time, return the right answer for any query time without overwriting history, without per-update LLM arbitration.

PART I – ATTENTION STEERING

Spectral Attention Steering for Prompt Highlighting

Waylon Li[1], Yuchen Niu[2], Yongxin Yang[4], Keshuang Li[3], Tiejun Ma[1], Shay B. Cohen[1]

University of Edinburgh | RayNeo | Huawei | Queen Mary University of London

ICLR 2026

Prompt highlighting as an attention control problem

Setting. A user (or upstream system) marks a subset of input tokens as *highlighted* so they should receive more attention from later tokens during generation.

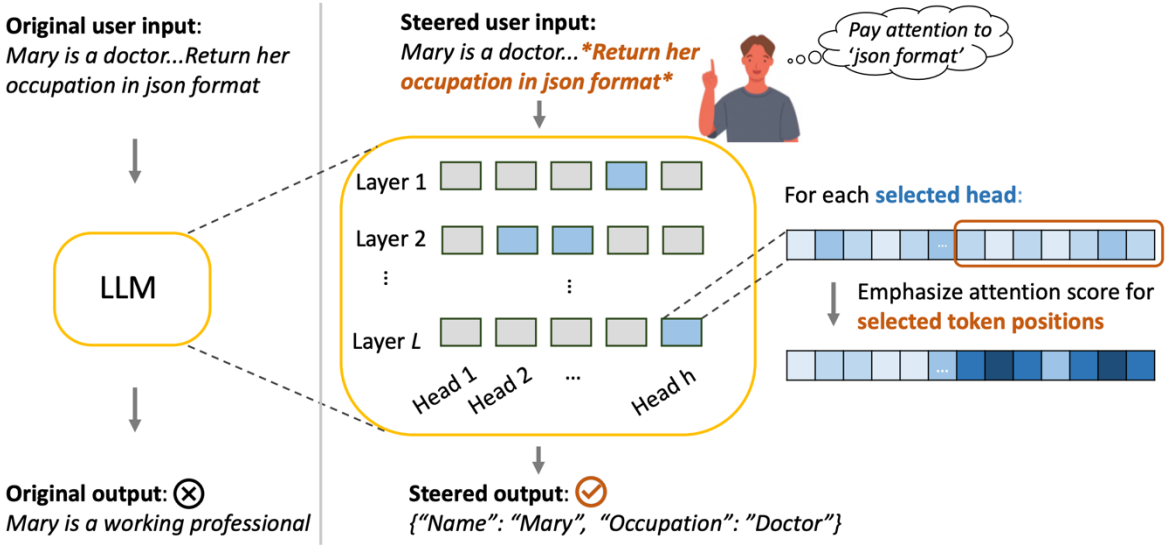
Examples. Knowledge-conflict editing · bias mitigation · instruction-following emphasis · long-context passage steering.

Representational framing. The control variable is *token relevance*. The representation that carries token identity into attention is the key vector.

The attention logits:

$$\boldsymbol{A}_{ij} = \frac{\boldsymbol{q}_i^\top \boldsymbol{k}_j}{\sqrt{d_k}}$$

Existing Method: PASTA



Zhang, Qingru, et al. "Tell your model where to attend: Post-hoc attention steering for llms." International Conference on Learning Representations. Vol. 2024. 2024.

The problem: post-hoc attention editing

Prompt highlighting steers LLM attention to user-specified tokens. Goal: amplify attention via additive bias $A'_{ij} = A_{ij} + \Delta_{ij}$

PASTA (Zhang et al., 2024) edits attention **after** computation: $[T(\boldsymbol{A})]_{ij} = \begin{cases} \alpha \dfrac{A_{ij}}{C_i}, & \text{if } j \in \mathcal{H}, \\ \dfrac{A_{ij}}{C_i}, & \text{otherwise.} \end{cases}$

This post-hoc manipulation causes:

- **FlashAttention incompatible** — must store full attention matrix
- **Costly head search** — task-specific per attention head
- **High overhead** — +1.03s latency, ~2× memory

Zhang, Qingru, et al. "Tell your model where to attend: Post-hoc attention steering for llms." International Conference on Learning Representations. Vol. 2024. 2024.

Relevance lives in key subspaces

We use contrastive prompt pairs (positive vs. negative for highlighted tokens), we extract key embeddings from shared token spans and visualise their pairwise shifts via PCA.

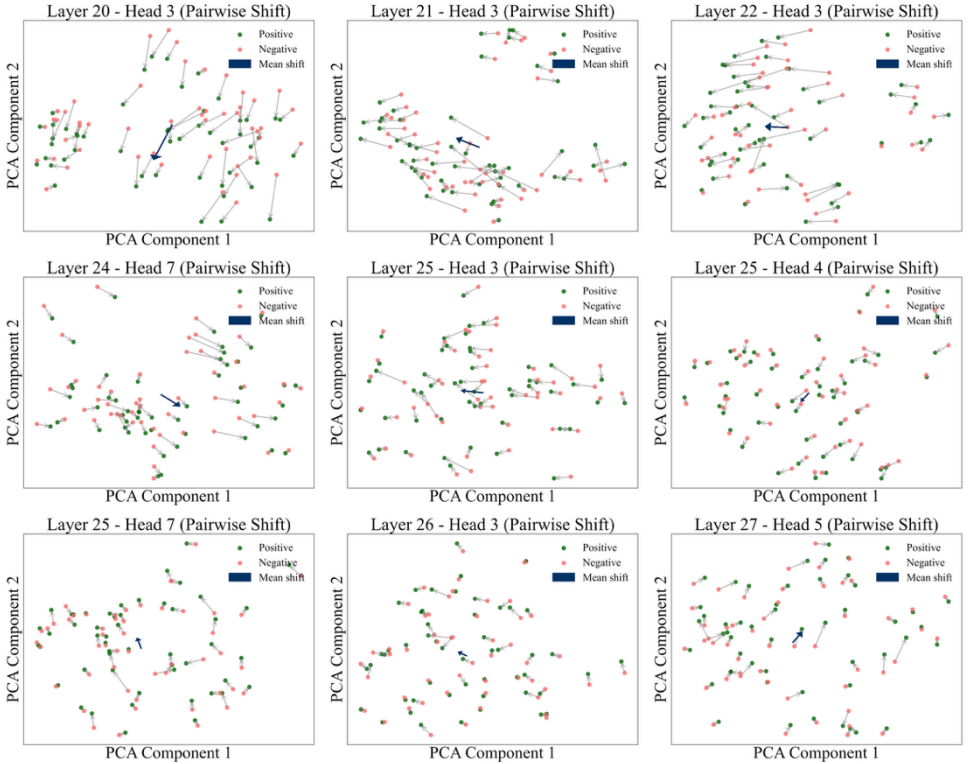
Group	Prompt
Neutral	Context: The portfolio manager allocates capital across equities and bonds.
Positive	Question: What does the portfolio manager allocate across equities and bonds? Context: The portfolio manager allocates capital across equities and bonds.
Negative	Question: What does the climate model simulate? Context: The portfolio manager allocates capital across equities and bonds.
Neutral	Context: The climate model simulates sea-level rise under different scenarios.
Positive	Question: What does the climate model simulate? Context: The climate model simulates sea-level rise under different scenarios.
Negative	Question: What does the portfolio manager allocate across equities and bonds? Context: The climate model simulates sea-level rise under different scenarios.

Relevance lives in key subspaces

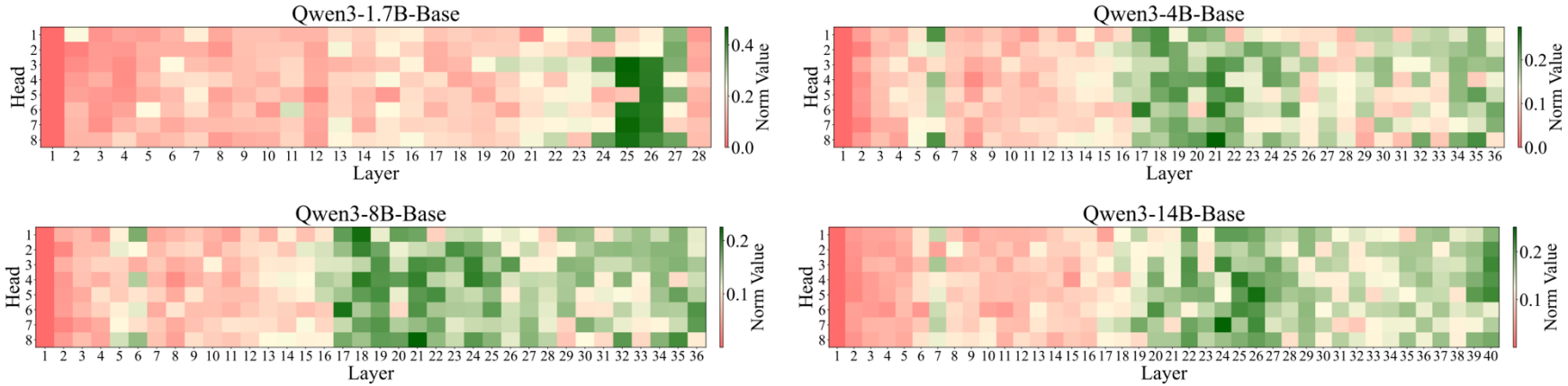
We use contrastive prompt pairs (positive vs. negative for highlighted tokens), we extract key embeddings from shared token spans and visualise their pairwise shifts via PCA.

Key Findings

- Certain (layer, head) pairs show **robust, consistent directional shifts** when token relevance changes
- Relevance is encoded in a **structured subspace** of key representations
- Grey arrows = individual shifts; dark blue arrow = mean displacement



Where in the LLM does the relevance live?



Key Observation

- Relevance sensitivity is **not uniform** across heads
- Large ℓ_2 shifts concentrate in **mid-to-late layers**
- Aligns with **retrieval heads** (Wu et al., 2025)

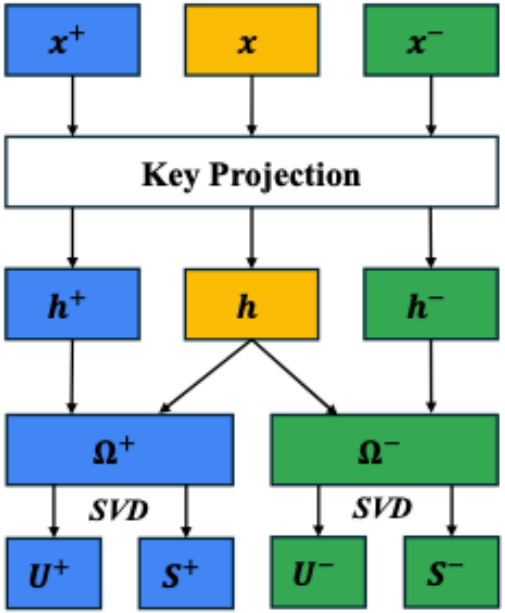
Wu, Wenhao, et al. "Retrieval head mechanistically explains long-context factuality." *International Conference on Learning Representations*. Vol. 2025. 2025.

Can we achieve the similar effect by editing Keys?

Method: Spectral Editing Key Amplification (SEKA)

Core Idea: Edit key embeddings ***before*** attention computation, not after.

- Use **spectral decomposition** (SVD) on contrastive key embeddings to learn a **relevance subspace**
- Amplify highlighted tokens via projection: $\boldsymbol{k}'_j = \boldsymbol{k}_j + g \cdot P \cdot \boldsymbol{k}_j$
- **Training-free**, FlashAttention compatible

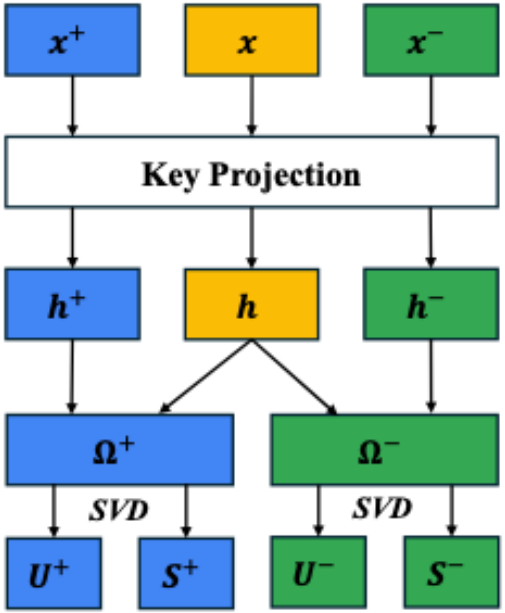


Method: Spectral Editing Key Amplification (SEKA)

Core Idea: Edit key embeddings ***before*** attention computation, not after.

- Use **spectral decomposition** (SVD) on contrastive key embeddings to learn a **relevance subspace**
- Amplify highlighted tokens via projection: $\boldsymbol{k}'_j = \boldsymbol{k}_j + g \cdot P \cdot \boldsymbol{k}_j$
- **Training-free**, FlashAttention compatible

$$\text{Logits}_{ij} = \frac{\boldsymbol{q}_i^\top \boldsymbol{k}_j}{\sqrt{d_k}} + \frac{\boldsymbol{q}_i^\top \left(\frac{g^+ \cdot \boldsymbol{P}^+_{\ell,h} \boldsymbol{k}_j + g^- \cdot \boldsymbol{P}^-_{\ell,h} \boldsymbol{k}_j}{2} \right)}{\sqrt{d_k}} = \boldsymbol{A}_{ij} + \boldsymbol{B}_{ij}$$

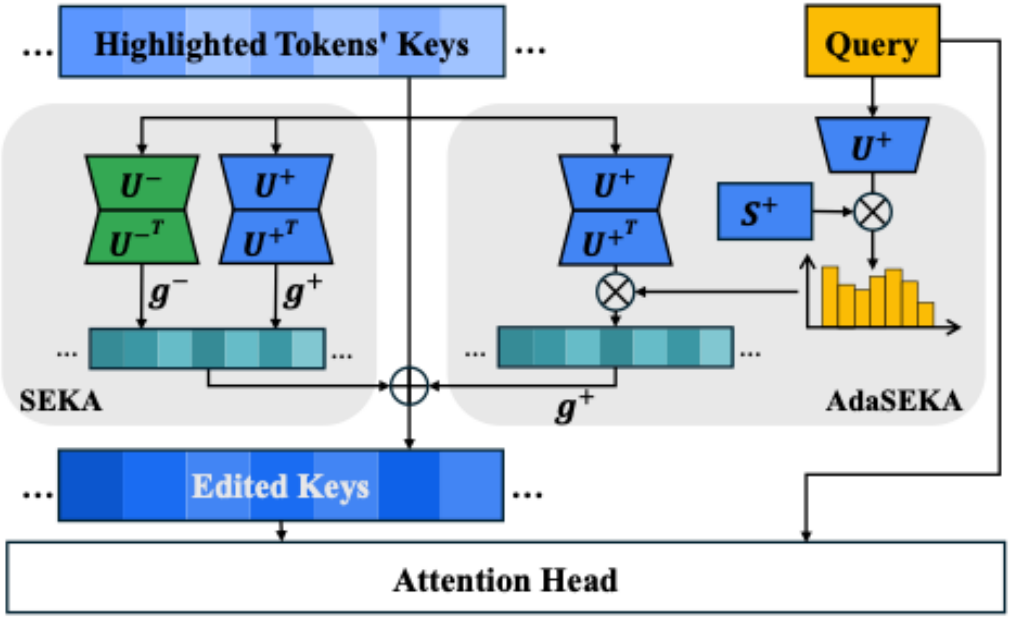


Adaptive SEKA – Query-adaptive composition of experts

Observation. Different tasks can have *different* relevance geometries.

AdaSEKA. Learn a small bank of expert projectors $\{P_e\}$ offline; at inference, route each query to a convex combination based on its alignment with each expert's singular directions.

$$k'_j = k_j + \sum_e w_e(q)\, P_e\, k_j$$



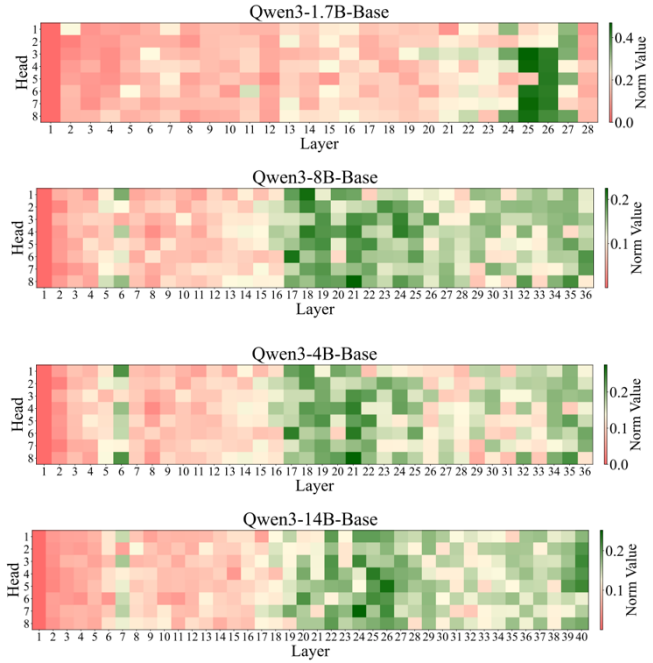
One remaining question: how shall we determine the head to apply steering on?

Head Selection Method

- Compute per-token ℓ_2 distance between h^+ and h^- for every (layer, head)
- Apply SEKA only where distance $> \delta_{min}$
- Avoids interfering with heads that don't encode relevance

$$D_{\ell,h} = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{h}^{+}_{\ell,h,i} - \boldsymbol{h}^{-}_{\ell,h,i}\right\|_2$$

$$D_{\ell,h} \geq \delta_{\min}$$



Benchmarks

Table 1: Summary of standard benchmarks for attention steering. **Tokens in bold** indicate where attention steering is applied.

Task	Prompts	Metrics
Counterfact	Previously, *[old fact]*. Currently, ***[new fact]***. *[question]*.	Efficacy score (ES), Paraphrase score (PS)
Bias in Bios	***[person's occupation]***. *[career history, may not directly related to prediction]*. *[person]* has the occupation of a/an ___	Accuracy (Acc.)
Pronouns changing	*[biographical contexts]*. **Substitute 'she' and 'he' with 'they' and generate the occupation of *[person]* after changing pronouns.**	Pronoun-weighted Lexical overlap Score (P. Score), All-changed P. Score

Results

Table 3: Inference overhead on Qwen3-8B-Base. Time is per-sample; memory is average peak usage.

Method	Avg. Time (s)	Peak Mem. (GB, B=10)	Peak Mem. (GB, B=1)
Original	0.55	27.63	16.72
PASTA	1.58 (+1.03)	50.75 (+23.12)	-
SPA	5.87 (+5.32)	-	17.71 (+0.99)
SEKA	0.58 (+0.03)	27.66 (+0.03)	16.75 (+0.03)
AdaSEKA	0.82 (+0.27)	43.22 (+15.59)	18.23 (+1.51)

Table 2: Performance on standard benchmarks. **Bold** = best. Underline = second best. We include two ablation studies for *SEKA*: *"w/o learn"* uses random projections instead of spectrally learned ones, and *"w/o learn&filt"* further removes the head filtering mechanism.

Model	Metric	Baselines				Our Methods			
		Original	**-marked	PASTA	SPA	*SEKA*	*w/o learn*	*w/o learn&filt*	*AdaSEKA*
Qwen3-4B	CounterFact (ES)	45.00	57.70	97.16	65.24	**99.02**	94.96	86.12	98.90
	CounterFact (PS)	45.64	52.12	96.03	57.71	98.61	92.38	86.20	**98.72**
	Bias in Bios (Acc.)	79.84	82.94	89.58	68.00	91.02	86.62	71.76	**91.86**
	Pronoun (P. Score)	93.14	95.76	**95.82**	80.27	95.18	90.42	41.98	94.54
	Pronoun (A. P. Score)	90.52	93.88	**94.64**	78.19	93.26	88.66	36.95	92.08
Qwen3-8B	CounterFact (ES)	39.04	56.24	92.70	69.26	**99.08**	96.12	95.18	99.00
	CounterFact (PS)	39.59	49.80	91.68	58.76	98.96	94.74	89.69	**98.97**
	Bias in Bios (Acc.)	76.08	80.60	86.32	37.02	**88.74**	87.26	74.90	88.50
	Pronoun (P. Score)	98.00	98.10	98.86	72.61	98.56	98.12	80.53	**99.68**
	Pronoun (A. P. Score)	97.84	97.84	98.72	74.95	98.26	97.90	80.85	**99.52**
Qwen3-14B	CounterFact (ES)	37.56	45.52	76.84	84.22	98.92	86.28	95.26	**99.00**
	CounterFact (PS)	36.12	40.12	66.33	76.11	99.02	88.07	92.02	**99.15**
	Bias in Bios (Acc.)	85.22	90.94	88.46	57.86	90.28	88.02	88.44	**91.22**
	Pronoun (P. Score)	98.42	98.86	90.98	91.60	98.66	96.32	88.60	**99.88**
	Pronoun (A. P. Score)	98.22	98.68	90.94	92.20	98.54	96.36	89.76	**99.86**
Gemma3-4B	CounterFact (ES)	55.04	57.56	78.36	93.90	98.04	95.14	94.46	**98.74**
	CounterFact (PS)	47.77	45.82	59.53	91.92	98.83	92.25	91.98	**99.05**
	Bias in Bios (Acc.)	89.90	91.00	82.58	48.02	92.42	85.60	77.16	**92.92**
	Pronoun (P. Score)	41.34	38.86	67.39	76.05	81.53	53.58	51.78	**93.76**
	Pronoun (A. P. Score)	35.25	32.45	66.43	74.45	81.11	48.82	51.94	**93.58**
Gemma3-12B	CounterFact (ES)	45.34	48.72	68.30	93.76	**98.86**	63.08	60.96	92.48
	CounterFact (PS)	37.21	36.69	71.72	91.24	**99.27**	50.59	76.37	93.65
	Bias in Bios (Acc.)	91.26	92.90	**94.72**	46.88	93.04	91.84	90.54	91.14
	Pronoun (P. Score)	93.92	95.78	68.47	86.41	**97.70**	47.26	55.56	96.88
	Pronoun (A. P. Score)	94.96	96.42	68.01	84.99	**97.24**	51.24	58.76	95.84

Takeaways

SEKA introduces a mathematically rigorous, training-free approach to direct an LLM's focus.

- **Fundamentally Efficient:** Intervenes on key embeddings, eliminating the N × N matrix materialisation bottleneck.
- **FlashAttention Compatible:** Preserves modern IO-aware optimisation.
- **Highly Performant:** Sets state-of-the-art on prompt highlighting and positional recall benchmarks with negligible memory/latency overhead.

Ge, Yuyao, et al. "Prism-Delta: Differential Subspace Steering for Prompt Highlighting in Large Language Models." *arXiv preprint arXiv:2603.10705* (2026).

PART II – Agentic Memory

Time is Not a Label: Continuous Phase Rotation for Temporal Knowledge Graphs and Agentic Memory

Waylon Li[1], Jiaxin Zhang[2], Jim Yang[3], Tiejun Ma[1], Ryman Guo[2]

University of Edinburgh | LIGHTSPEED | University of St Andrews

Preprint

Why graph-based memory needs a model of time

Not every fact is equally permanent. *Time* is not a metadata column but part of the *truth condition* of a relation.

STATIC FACT

(Obama, born_in, Hawaii)

Should remain valid for all query time. The relation born_in has no temporal volatility.
A memory that overwrites or down-ranks old facts on recency will quietly lose this.

DYNAMIC FACT

(Obama, president_of, USA)

Valid only inside a specific time window. The relation president_of is highly volatile.
A memory that treats all relations identically will conflate this with the current president.

Existing workarounds for graph-based agentic memory

APPROACH A

Destructive overwriting

When a new fact contradicts an old one, delete or overwrite the old fact in place.

Cost. Loses the historical record entirely. Cannot answer who was president in 2010? After Trump's term is recorded.

APPROACH B

LLM arbitration at ingestion

For every incoming fact, call an LLM to decide whether and how to update the store relative to existing facts.

Cost. Per-ingestion LLM call. Expensive at scale.

APPROACH C

Recency sorting at retrieval

At query time, re-rank candidate facts by timestamp recency relative to the query.

Cost. Treat all relations identically. Helps dynamic relations; quietly destroys static ones.

Where do we go for a memory that *does* represent time?

Every workaround on the previous slide is a post-hoc fix wired around a memory that has *no internal representation of temporal validity*.

But actually, the community has spent roughly a decade on *exactly this question* regarding temporal knowledge graph embedding (TKGE) learning.

Type	Representative methods	What they do
1. Additive / projection-based time conditioning	**T-TransE, HyTE**	Add a time vector, or project facts into time-specific spaces.
2. Non-rotation temporal factorisation / embedding models	**DE-SimplE**: Goel et al., 2020; **TComplEx**: Lacroix et al., 2020; **TLT-KGE**: Zhang et al., 2022; **HGE**: Pan et al., 2024; **TimeGate**: Shen et al., 2025	Encode temporal facts through factorisation, diachronic embeddings, geometric product spaces, or attention/gating.
3. Discrete geometric / rotation-based TKGE	**RotatE**; **TeRo**; **ChronoR**; **RotateQVS**; **TeAST**; **TCompoundE**; **3DG-TE**	Use rotations or geometric transformations to model temporal evolution.

- Time stays as a discrete label
- No learned per-relation volatility

Why rotation? Temporal validity as a learned geometric operator

Among the TKGE families on the previous slides, only the rotational view gives us the geometric handle we need.

Time-invariant rotation (RotatE). Each relation is a per-dimension complex unit $\boldsymbol{\rho_r} = \mathrm{e}^{i\theta_r}$.

$$s(h,r,t) = \gamma - ||h \circ \rho_r - t||$$

Discrete-time rotation (ChronoR).

$$\mathrm{g}(h,r,t,\tau) = \langle h \circ [r|\tau] \circ r_2, t \rangle$$

$[r|\tau]$ is the concatenated relation-time embedding while τ is the discrete time embedding.

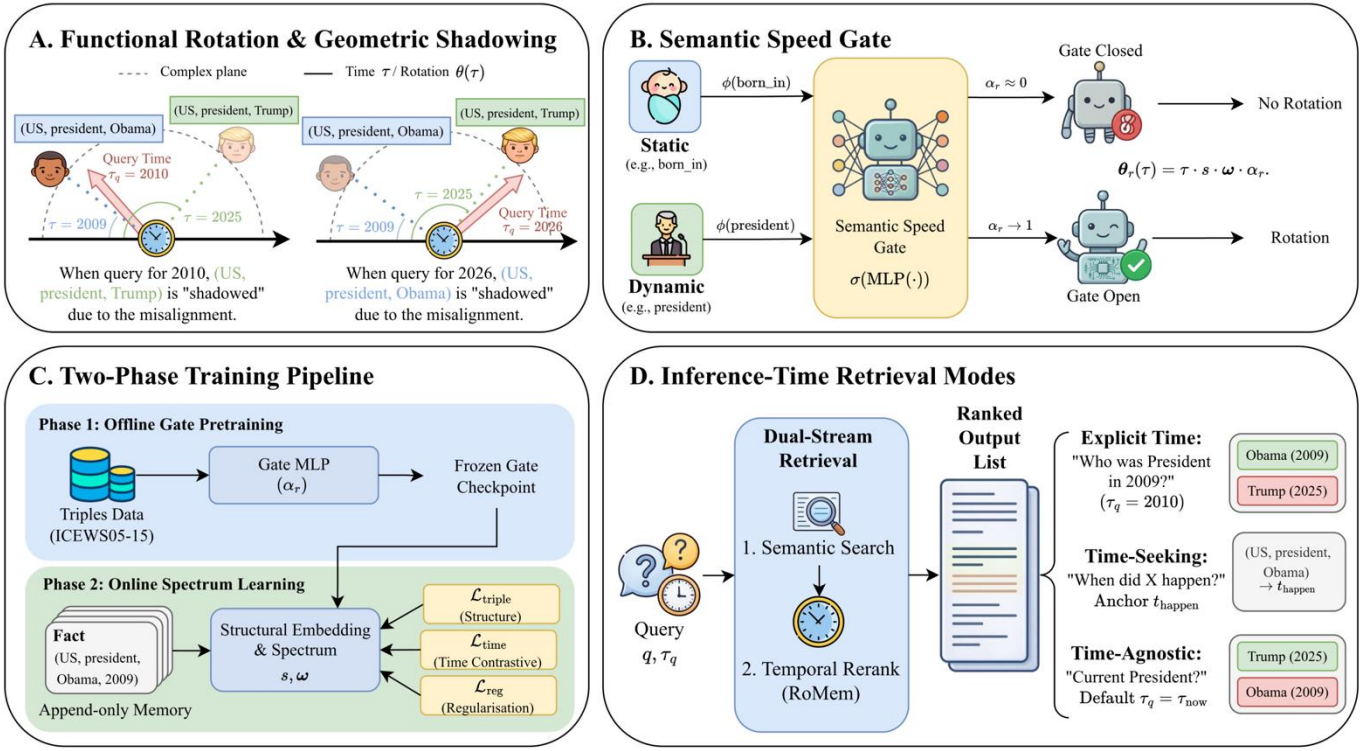
1. Continuity over time. Model $\theta_r(\tau)$ as a continuous function, not a lookup so that we can interpolate between observed.

2. **A generalisable per-relation velocity.** Factor the angle as

$$\theta_r(\tau) = s \cdot \alpha_r \cdot \tau \cdot \omega$$

With s as the global time scale, α_r is the semantic speed gate, τ is the continuous timestamp and ω is the frequency spectrum (learnable).

RoMem


A. Functional Rotation & Geometric Shadowing
Complex plane
Time τ / Rotation $\theta(\tau)$
(US, president, Obama)
(US, president, Trump)
Query Time $\tau_q = 2010$
$\tau = 2009$
$\tau = 2025$
When query for 2010, (US, president, Trump) is "shadowed" due to the misalignment.
(US, president, Obama)
(US, president, Trump)
$\tau = 2025$
$\tau = 2009$
Query Time $\tau_q = 2026$
When query for 2026, (US, president, Obama) is "shadowed" due to the misalignment.
B. Semantic Speed Gate
Static
(e.g., born_in)
ϕ(born_in)
Dynamic
(e.g., president)
ϕ(president)
Semantic Speed Gate
σ(MLP(·))
$\alpha_r \approx 0$
$\alpha_r \to 1$
Gate Closed
No Rotation
$\theta_r(\tau) = \tau \cdot s \cdot \omega \cdot \alpha_r$.
Rotation
Gate Open
C. Two-Phase Training Pipeline
Phase 1: Offline Gate Pretraining
Triples Data (ICEWS05-15)
Gate MLP (α_r)
Frozen Gate Checkpoint
Phase 2: Online Spectrum Learning
Fact
(US, president, Obama, 2009)
Append-only Memory
Structural Embedding & Spectrum
s, ω
$\mathcal{L}_{\text{triple}}$ (Structure)
$\mathcal{L}_{\text{time}}$ (Time Contrastive)
$\mathcal{L}_{\text{reg}}$ (Regularisation)
D. Inference-Time Retrieval Modes
Query
q, τ_q
Dual-Stream Retrieval
1. Semantic Search
2. Temporal Rerank (RoMem)
Ranked Output List
Explicit Time:
"Who was President in 2009?"
($\tau_q = 2010$)
Obama (2009)
Trump (2025)
Time-Seeking:
"When did X happen?"
Anchor t_{happen}
(US, president, Obama) $\to t_{\text{happen}}$
Time-Agnostic:
"Current President?"
Default $\tau_q = \tau_{\text{now}}$
Trump (2025)
Obama (2009)

Continuous phase rotation & Semantic speed gate

$$\boldsymbol{\theta}_r(\tau) = s \cdot \alpha_r \cdot \tau \cdot \boldsymbol{\omega}$$

$$\alpha_r = \sigma\big(\mathrm{MLP}(\phi(r))\big)$$

- **Static relations.** The gate outputs small α_r. The rotation is essentially the identity where the embeddings sit on the unit circle and stay there as time passes.
- **Dynamic relations.** The gate outputs large α_r (close to 1). The fact rotates out of phase as time advances and back into phase when a query asks about that fact's actual window.
- **Continuous.** No buckets, no discrete timestamp projections so we have a smooth functional operator on the time.

- **Static-looking phrases.** 'born_in' / 'authored' -> low speed -> less rotation
- **Dynamic-looking phrases.** 'president_of' / 'CEO_of' / 'eat' -> high speed -> rotation kicks in.
- **Pretrained, then frozen.** The gate is trained on large temporal knowledge graph dataset such as ICEWS05-15, then frozen during the agentic memory ingestion phase. We hypothesize the rotation speed can be generalise across relations with similar semantic meaning.

Scoring function

$$\mathbf{v}_r^c(\mathbf{e}, \tau) = \mathrm{Rot}(\mathbf{e}^c, \boldsymbol{\theta}_r(\tau))$$

$$\tilde{\mathbf{v}}_r^c(\mathbf{e}, \tau) = \mathbf{v}_r^c(\mathbf{e}, \tau) \odot \mathbf{w}_r^c \odot \hat{\mathbf{w}}_r^c$$

$$s_{\mathrm{kge}}\big((h, r, t) \mid \tau\big) = \sum_{c=1}^{k} \big\langle \tilde{\mathbf{v}}_r^c(\mathbf{e}_h, \tau),\ \mathbf{v}_r^c(\mathbf{e}_t, \tau) \big\rangle$$

Benchmark

Track 1 | TKGE link prediction

ICEWS05-15: political events, day resolution

Metrics: MRR / Hit@1/ Hit@3 / Hit@10

Baselines:

- Non-rotation: DistMult, DE-SimplE, TComplEx, HGE, TimeGate
- Rotation: TeRo, ChronoR, RotateQVS, TeAST, TCompoundE, 3DG-TE

Track 2 | Agentic Memory

MultiTQ: heavy temporal QA
LoCoMo: mixed general / temporal long conversations
DMR-MSC: purely general QA
FinTMMBench: zero-shot to financial relations

Baselines: Zep, Mem0, A-Mem, LicoMemory, HippoRAG 2

Track 1 Results

Table 1: Results on ICEWS05-15. Baseline results are taken from Li et al. (2025) and Shen et al. (2025). Best results are in **bold**. Green cells indicate results where RoMem improves over its backbones (DistMult and ChronoR).

Method	MRR	Hit@1	Hit@3	Hit@10
Non-Rotation Based				
DistMult (2015)	45.6	33.7	-	69.1
DE-SimplE (2020)	51.3	39.2	57.8	74.8
TComplEx (2020)	66.5	58.3	71.6	81.1
TLT-KGE (2022)	68.6	60.7	73.5	83.1
HGE (2024a)	68.8	60.8	74.0	83.5
TimeGate (2025)	69.2	61.3	74.5	83.7
Rotation Based				
TeRo (2020)	58.6	46.9	66.8	79.5
ChronoR (2021a)	68.4	61.1	73.0	82.1
RotateQVS (2022)	63.3	52.9	70.9	81.3
TeAST (2023)	68.3	60.4	73.2	82.9
TCompoundE (2024)	69.2	61.2	74.3	83.7
3DG-TE (2025)	69.4	61.4	74.7	**84.1**
RoMem (Ours)				
RoMem-DistMult	62.1	54.2	66.3	77.2
RoMem-ChronoR	**72.6**	**66.8**	**75.9**	83.7

† We use $k = 3$ for (RoMem-)ChronoR, following Sadeghian et al. (2021a). k is the rotation dimensionality defined therein.

Track 2 Results

Table 2: Comprehensive evaluation of RoMem. **(a) MultiTQ:** Heavy temporal reasoning. **(b) LoCoMo:** Hybrid reasoning (Recall@10). **(c) DMR-MSC:** Static memory preservation. **(d) FinTMMBench:** Zero-shot domain generalisation. Implementation = LLM for graph construction (named entity recognition and triple extraction) + Embedding model. Best results are in **bold**. Green cells indicate results where RoMem improves over its HippoRAG backbone.

(a) MultiTQ (RQ2, Heavy Temporal)

Method	MRR	Hit@3	Hit@10	Acc@5	Acc@10
GPT-5-mini + text-embedding-3-small					
Zep	0.192	0.208	0.310	0.110	0.118
Mem0	0.174	0.190	0.282	0.122	0.122
A-Mem[1]	–	–	–	–	–
LicoMem.	0.149	0.160	0.292	0.114	0.128
HippoRAG	0.203	0.232	0.348	0.112	0.102
RoMem	**0.337**	**0.384**	**0.502**	**0.366**	**0.392**
LLaMA-3.1-70B + BGE-M3					
Zep	0.217	0.252	0.370	0.098	0.116
Mem0	0.228	0.264	0.356	0.120	0.114
A-Mem[1]	–	–	–	–	–
LicoMem.	0.159	0.182	0.304	0.114	0.120
HippoRAG	0.236	0.266	0.354	0.122	0.116
RoMem	**0.316**	**0.342**	**0.440**	**0.312**	**0.338**

(b) LoCoMo (RQ2, Hybrid Tasks)

Method	Single Hop	Multi Hop	Open Domain	Temporal Reason	Average
GPT-5-mini + text-embedding-3-small					
Zep	0.557	**0.861**	0.831	0.553	0.770
Mem0	0.740	0.832	0.883	0.690	0.834
A-Mem	0.740	0.846	0.860	0.691	0.825
LicoMem.	0.727	0.856	0.848	0.661	0.816
HippoRAG	0.711	0.837	0.862	0.645	0.815
RoMem	**0.768**	0.850	**0.904**	**0.726**	**0.857**
Implementation: LLaMA-3.1-70B + BGE-M3					
Zep	0.557	**0.861**	0.831	0.553	0.770
Mem0	0.746	0.860	0.875	0.737	**0.839**
A-Mem	0.658	0.776	0.777	0.702	0.750
LicoMem.	0.605	0.768	0.725	0.584	0.703
HippoRAG	0.717	0.852	0.870	0.732	0.830
RoMem	**0.759**	0.824	**0.879**	**0.759**	0.838

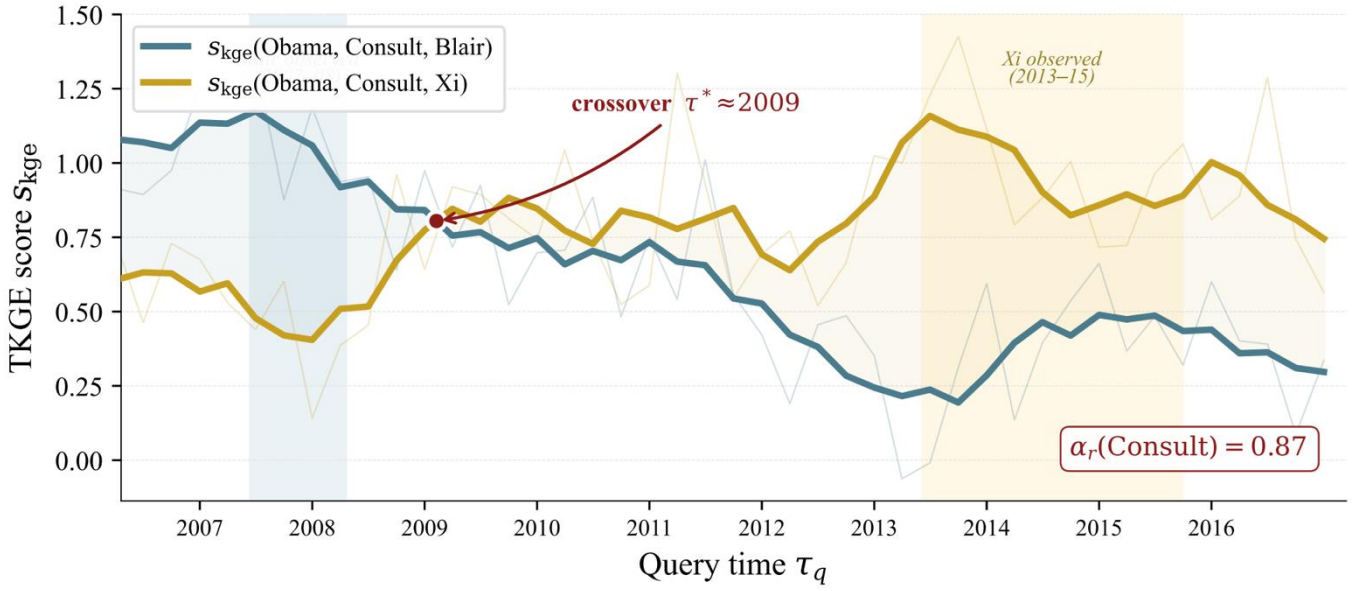
(c) DMR-MSC (RQ2, Static Memory)

Method	MRR	Hit@1	Hit@3	Acc@5	Acc@10
GPT-5-mini + text-embedding-3-small					
Zep	0.170	0.110	0.180	0.302	0.376
Mem0	0.847	0.766	0.926	0.858	0.848
A-Mem	0.825	0.732	0.912	0.848	0.856
LicoMem.	0.326	0.224	0.372	0.670	0.728
HippoRAG	0.848	0.768	0.926	0.852	0.850
RoMem	**0.856**	**0.774**	**0.934**	**0.862**	**0.858**
Implementation: LLaMA-3.1-70B + BGE-M3					
Zep	0.333	0.232	0.394	0.384	0.428
Mem0	0.821	0.714	0.926	0.758	0.770
A-Mem	0.823	0.732	0.902	0.728	0.738
LicoMem.	0.202	0.138	0.228	0.258	0.338
HippoRAG	0.818	0.718	0.912	0.768	0.776
RoMem	**0.847**	**0.760**	**0.930**	**0.774**	**0.786**

(d) FinTMMBench (RQ3)

Method	MRR	R@5	R@10	Acc@5	Acc@10
GPT-5-mini + text-embedding-3-small					
Zep	0.703	0.644	0.759	0.480	0.520
Mem0	0.691	0.645	0.768	0.550	0.610
A-Mem	0.716	0.647	0.796	0.540	0.640
LicoMem.	0.488	0.480	0.609	0.480	0.590
HippoRAG	0.690	0.645	0.768	0.550	**0.650**
RoMem	**0.728**	**0.673**	**0.779**	**0.580**	**0.650**
Implementation: LLaMA-3.1-70B + BGE-M3					
Zep	0.515	0.510	0.591	0.430	0.450
Mem0	0.718	0.647	0.765	0.570	0.610
A-Mem	0.650	0.631	0.742	0.520	0.590
LicoMem.	0.554	0.559	0.662	0.460	0.520
HippoRAG	0.724	0.680	0.766	0.610	0.610
RoMem	**0.726**	**0.707**	**0.793**	**0.620**	**0.650**

Case study 1 - Geometric shadowing of obsolete facts



Case study 2 – Semantic Speed Gate values

Table 5: Pretrained semantic speed gate values. Higher α_r = faster rotation. *Seen*: appeared in ICEWS05-15 during pretraining; *Unseen*: zero-shot via text embedding similarity.

Relation	α_r	**Category**
Seen during gate pretraining (ICEWS05-15)		
Consult	0.87	Dynamic
Host a visit	0.86	Dynamic
Engage in negotiation	0.63	Dynamic
Sign formal agreement	0.53	Dynamic
Cooperate economically	0.16	Static
Cooperate militarily	0.09	Static
Unseen (zero-shot via text embeddings)		
met with	0.71	Dynamic
visited	0.64	Dynamic
negotiated with	0.62	Dynamic
CEO of	0.44	Moderate
capital of	0.36	Moderate
species	0.22	Static
citizen of	0.17	Static

Takeaways

- Append-only memory can still resolve contradictions by representational learning.
- Temporal validity can be encoded as geometry.
- Static and dynamic facts need to be handled differently for external knowledge base.

Same recipe, different applications

Paper	Latent variable	Computational space	Compact geometry	Operator
SEKA	Token relevance	Transformer key embeddings	Low-rank subspace per (layer, head)	Projection
RoMem	Temporal validity	KG entity / time embeddings	Phase angle with relation-conditioned speed	Rotation

Post-hoc → Representational learning:

- Post-hoc attention editing → key editing
- Post-hoc recency ranking / explicit LLM management → rotation-based embedding learning

Thanks!

SEKA:

- Paper: https://arxiv.org/abs/2603.01281
- Codes: https://github.com/waylonli/SEKA
- Dataset: https://huggingface.co/datasets/waylonli/SEKA-datasets

RoMem:

- Paper: https://arxiv.org/abs/2604.11544
- Codes: https://github.com/Tencent/RoMem