GNNs for Technology Monitoring

How much graph structure helps predict emerging technologies?

Paul Bagourd

CYD Campus / Armasuisse
with Julian Jang-Jaccard

18 March 2026

Motivation

- Scientific output is growing faster than analysts can track manually.
- Use case: cybersecurity and space technologies evolve quickly and unevenly.

Question

How can we help analysts identify which concepts are most likely to gain interest in the coming years?

- Fact: science is a collaborative process.

Research question

Once the target emergence score to forecast is defined carefully, do graph-temporal models outperform strong node-wise temporal baselines?

Example of concepts likely to gain momentum (Top 25)

Raw-view (NoGraph infer, October 2027)

Rank	Keyword	Emergence score
1	leo satellite	27.818
2	navigation satellite	16.426
3	real time	10.161
4	satellite networks	9.952
5	global navigation satellite systems	6.545
6	point positioning	5.724
7	multiple access	5.110
8	resource allocation	4.803
9	positioning accuracy	4.756
10	satellite terrestrial	4.669
11	inter satellite	4.556
12	ntn	4.358
13	high precision	4.299
14	optimization problem	4.274
15	large scale	3.757
16	deep reinforcement learning	3.539
17	neural networks	3.379
18	low latency	3.307
19	gnss receiver	3.065
20	data transmission	3.046
21	satellite links	2.991
22	noise ratio	2.980
23	channel state	2.978
24	space air ground	2.966
25	noma	2.885

Forecasting as Uncertainty Reduction

- Prediction relies on repeatable structure in past observations.
- Better inputs or inductive bias should reduce forecast uncertainty.
- Hypothesis: collaboration and co-occurrence structure may carry predictive signal.

In Shannon terms

$X_{\leq t}$: observable history up to month t

$Y_{t+\Delta}$: future technological state

The world induces an unknown joint distribution

$$p(Y_{t+\Delta}, X_{\leq t})$$

A forecasting model learns a surrogate distribution q_θ.

$$H(Y_{t+\Delta} \mid S_t^{\text{full}}) \leq H(Y_{t+\Delta} \mid X_{\leq t}) \leq \mathbb{E}_p[-\log q_\theta(Y_{t+\Delta} \mid X_{\leq t})]$$

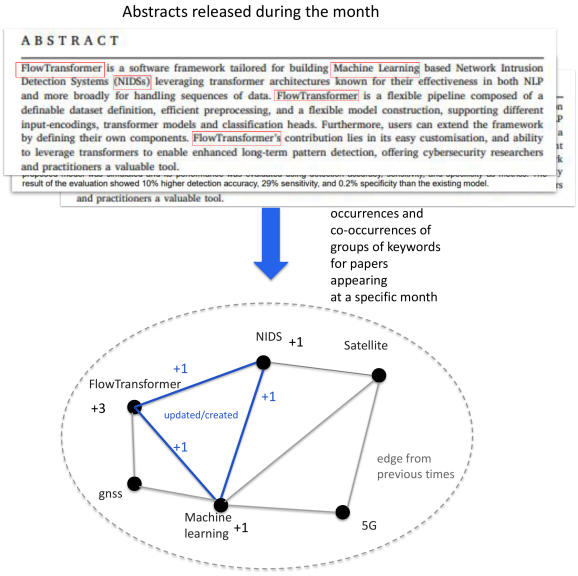
Objective: minimise empirical negative log-likelihood (cross-entropy)

Pipeline Overview

1. Data collection and preprocessing
2. Keyword extraction and curation
3. Temporal graph construction and feature engineering
4. Forecasting with graph neural networks

Core modeling idea

Build monthly keyword co-occurrence graphs, derive node features, then forecast a composite emergence score 24 months ahead.

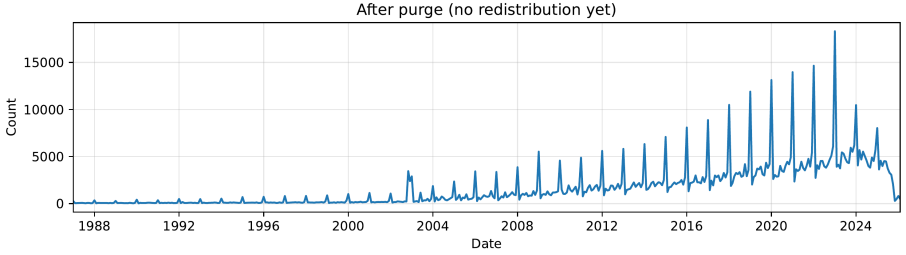


Stage 1: Corpus Construction

- Source: OpenAlex bibliographic corpus.
- Topic filtering isolates the cybersecurity-in-space use case.
- Deduplication, metadata cleaning, and abstract checks remove noisy records.
- Temporal corrections redistribute ingestion spikes and reduce recent-tail artefacts.

Corpus summary

- Initial crawl: 1,2M records
- After filtering and abstract checks: 780k records (topics union)
- Final corpus: 43k papers (cybersecurity and space technology intersection)



Stage 2: Keyword Extraction and Curation

Extraction

- KeyBERT + SPECTER on 15,000 abstracts
- About 190k unique bigram and trigram extracted

Curation

- Stop-list filtering removes boilerplate phrases ("http www", "xlink gt")
- Forced keywords keep niche but important concepts ("qkd")
- Final graph scope: top 500 keywords + 29 forced terms

Merging

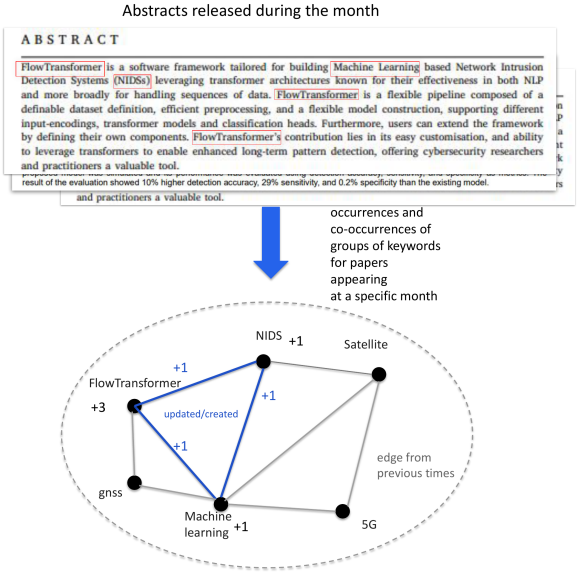
Syntactic merging allowed a priori: ["neural network", "neural networks", "neural net"]
Semantic merging allowed only a posteriori: ["leo satellites","leo satellites network"]

Stage 3: Temporal Graph Construction and Node Features

- Each node is a keyword/concept
- An edge links two keywords when they appear in the same abstract during the same month.
- Occurrences are counted at token level; co-occurrences are counted at document level.
- Compute one adjacency matrix per month

Monthly graph construction

Build one weighted co-occurrence graph per month from the abstracts released during that month.



Stage 3: Temporal Graph and Node Features

3 node/keyword/concept features

- Occurrences
- Total co-occurrences (sum of edges)
- Reference features

We build a causal reference-flow feature from outgoing references. Each month, a concept receives reference credit when newly published papers cite earlier in-corpus papers associated with that concept.

$$x_{c,t}^{\text{flow,frac_split}} = \sum_{p:\, m(p)=t} \sum_{r \in C(p)} \frac{1}{|R(p)|} \frac{1}{|C(r)|} \mathbf{1}[c \in C(r)],$$

Interpretation: each month, a new citing paper distributes credit to keywords of referenced in-corpus papers with weight

$$\frac{1}{|R(p)|} \cdot \frac{1}{|C(r)|},$$

then the monthly flow is cumulatively aggregated over time.

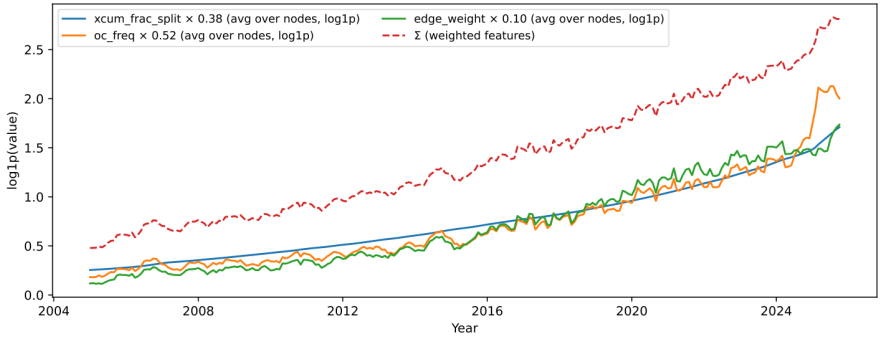
Stage 3: Temporal Graph and Node Features

Emergence score $E(\mathit{concept}, \mathit{time})$

Balanced weighted sum of the three features:

$$E(c,t) = w_{\mathrm{ref}}\, x_{\mathrm{ref}}(c,t) + w_{\mathrm{occ}}\, x_{\mathrm{occ}}(c,t) + w_{\mathrm{edge}}\, x_{\mathrm{edge}}(c,t)$$

$$FW = [0.3831,\ 0.5189,\ 0.0980]$$



Target Views

Raw gain

$$y_{c,t}^{\mathrm{res}} = E(c, t+\Delta) - E(c, t)$$

Interpretation: predict future mass dominance.

Smoothed relative growth

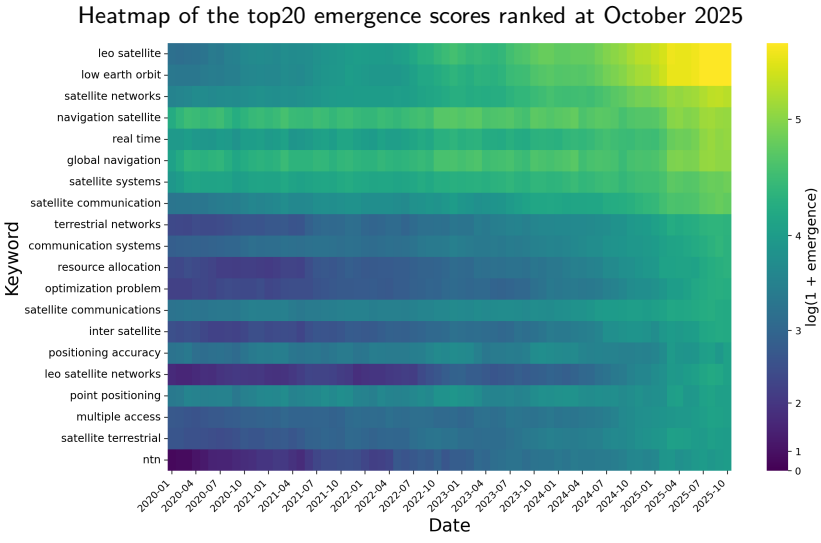
$$\widetilde{E}(c,t) = \frac{1}{12}\sum_{k=0}^{11} E(c, t-k)$$

$$R(c,t) = \frac{\widetilde{E}(c,t) - \widetilde{E}(c,t-12)}{\widetilde{E}(c,t-12) + \tau}$$

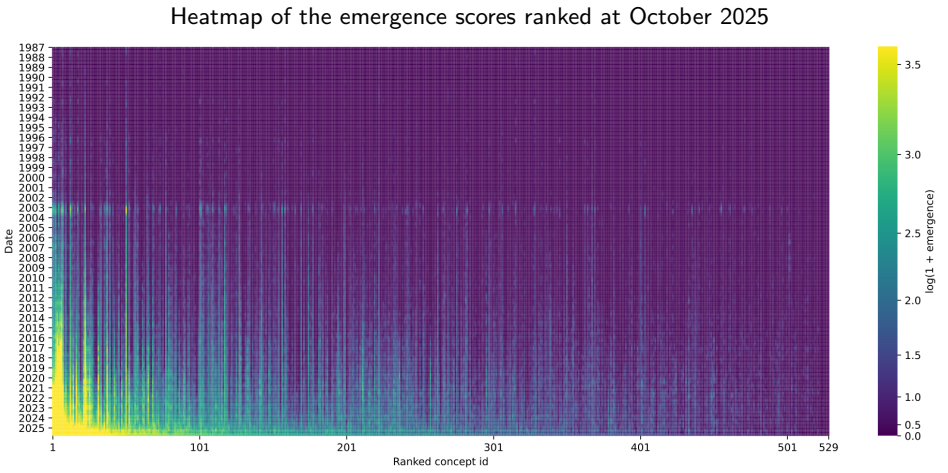
$$y_{c,t}^{\mathrm{rel}} = R(c, t+\Delta)$$

Here τ is an additive stabilizer (set from a low quantile of lagged positive values) to avoid unstable ratios when $\widetilde{E}(c, t-12)$ is close to zero.
Interpretation: predict relative momentum, favor niche accelarating concepts.

Observed Signal

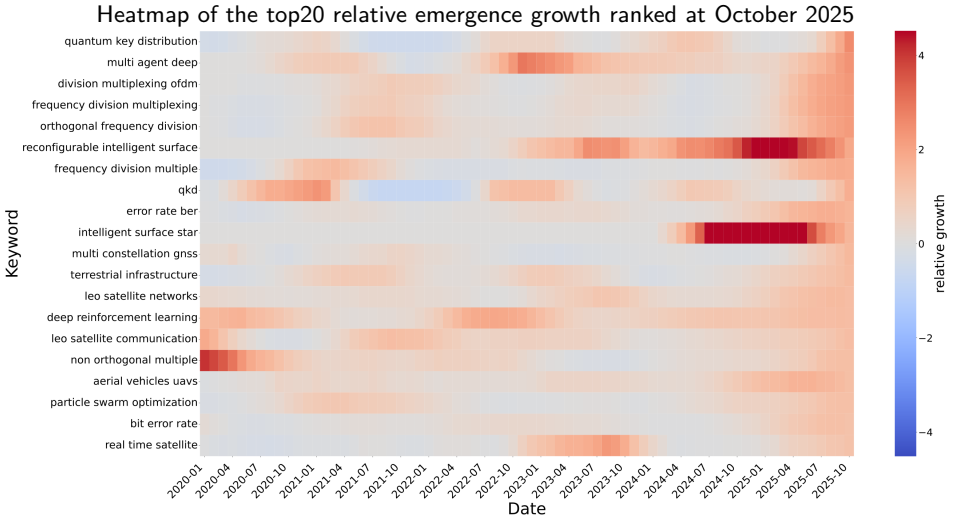


Observed Signal

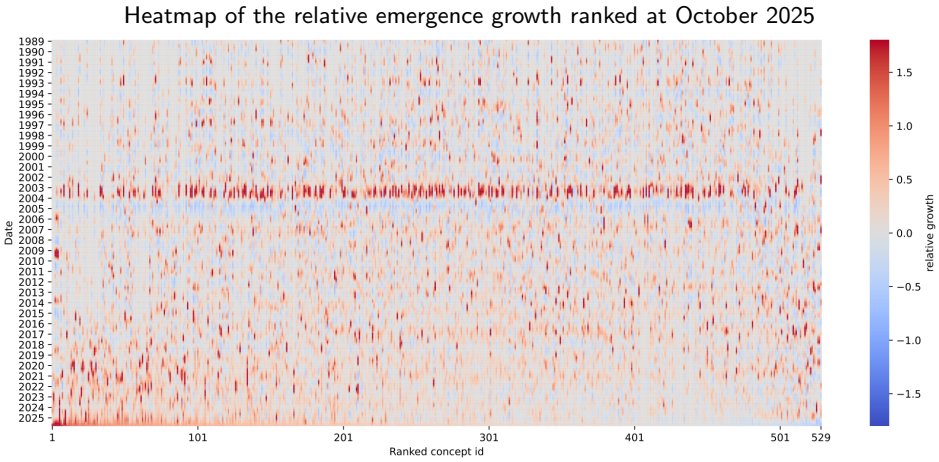


- The useful modeling regime starts around 2005; earlier years are mostly quiet.

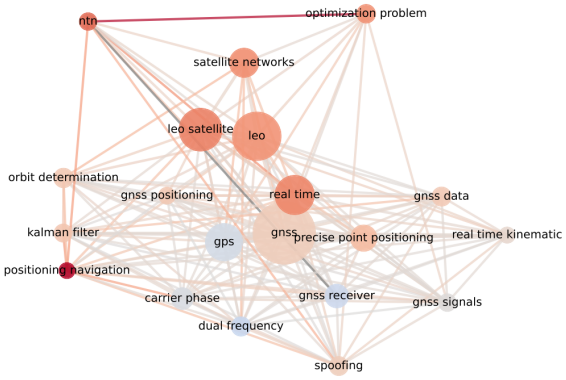
Observed Signal



Observed Signals



Recent Dynamics: Analyst View



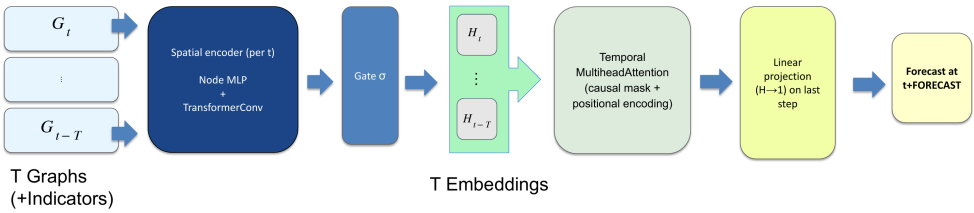
Top20 most emerging concepts in October 2025.

- Bubble size: current emergence score.
- Link distances: co-occurrence structure.
- Color: recent relative growth.

Forecasting Model: GTAN Base

Architecture flow

- Input: graphs + node features
- Node MLP encoder: build per-keyword embeddings.
- Graph layer: exchange information through TransformerConv each month.
- Temporal layer: attend over recent months before the regression head.
- Output: one forecast per node.



Models Compared

Persistence:	predicts no change from the current state ($\hat{x}_{t+\Delta} = x_t$)
Drift:	extrapolates recent node-wise linear trend forward and ($\hat{x}_{t+\Delta} = x_t + \Delta \cdot \frac{x_t - x_{t-T}}{T}$).
NoGraph:	temporal model using only node histories, no message passing
Graph_base:	graph-temporal model using co-occurrence structure
Graph_mix:	learned node/graph blend at output head
Graph_multi:	multi-lag edges + auxiliary horizons
Graph_multi_mix:	multi-lag + learned node/graph blend

- Roughly 6k parameters per learned models
- Baselines test how much can be explained by inertia and trend alone
- Node-only vs graph models test whether co-occurrence structure adds predictive value

Training Protocol

Training loss

We choose RMSE loss: more sensitive to large misses than MAE

Temporal setup

- Input window: 12 months
- Forecast horizon: 24 months
- Sliding windows: 215 total
- Split: 129 train, 43 validation, 43 test

Scale

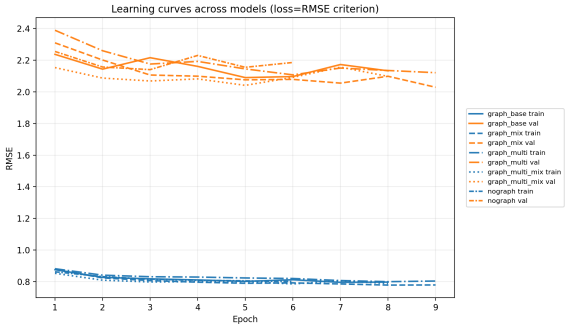
- 55,563 train targets
- 20,730 validation targets
- 21,394 test targets
- About 443 active nodes per month

Calendar split



Overfitting vs Regime Shift

- There is some classic overfitting signal: training loss is consistently much lower than validation loss.



Raw-view training: train/validation curves for the learned models.

- But the bigger issue is that test is harder/different (regime shift).

Main Results

Model	Raw gain RMSE	Raw variance explained	Relative gain RMSE	Relative variance explained
graph_base	9.688	19.9%	0.329	-8.2%
graph_mix	9.752	18.9%	0.326	-6.2%
graph_multi	9.827	17.6%	0.327	-6.8%
graph_multi_mix	9.727	19.3%	0.325	-5.7%
nograph	9.871	16.9%	0.336	-13.1%
persist	11.207	-7.1%	0.394	-55.5%
drift	8.708	35.3%	0.675	< 0

- Raw gain is dominated by simple drift: most predictable structure is smooth trend and scale. Drift model likely better understand the regime shifts. Graph signal is not significantly helpful.
- Relative gain is learnable relative to practical baselines, but graph signal is still not significantly helpful.

Drift-Corrected Diagnostic

Drift-corrected diagnostic

$$y_{c,t}^{\text{drift}} = E(c,t) - \hat{E}^{\text{drift}}(c, t \mid t-\Delta)$$

Interpretation: predict deviation relative to the linear trend, highlight surprises beyond drift.

But

The best score comes from the node-only control, and the remaining signal is not clearly graph-specific.

Model	Raw gain RMSE	Drift-residual RMSE
Graph_base	9.688	8.551
Graph_mix	9.752	8.545
Graph_multi	9.827	8.538
Graph_multi_mix	9.727	8.560
NoGraph	9.871	8.532
Persist	11.207	11.207
Drift baseline	8.708	8.708

Horizon Sensitivity

	Drift-corrected				Relative			
Model	1Y	2Y	3Y	4Y	1Y	2Y	3Y	4Y
drift	3.429	9.291	13.166	18.751	0.464	0.622	0.805	1.052
graph_base	3.368	8.728	12.680	17.951	0.259	0.313	0.332	0.378
graph_mix	3.372	8.675	12.540	17.825	0.265	0.319	0.327	0.378
graph_multi	3.400	8.932	12.580	18.100	0.271	0.311	0.327	0.373
graph_multi_mix	3.695	8.868	12.286	18.094	0.265	0.306	0.329	0.376
nograph	3.401	8.709	12.096	17.754	0.264	0.330	0.328	0.378
persist	4.658	13.117	17.940	23.870	0.345	0.360	0.369	0.434

RMSE by horizon (1Y–4Y) under two target views. Best value in each column in orange.

Takeaways

- The dominant short-term signal in this dataset is trend-like rather than relational.
- Raw gain favors drift, while smoothed relative growth reveals a learnable niche-sensitive signal.
- Graph structure helps a little, but not enough to claim a strong advantage over a node-only temporal model.
- Likely reason is target-graph mismatch: the target mixes occurrence, co-occurrence, and citation effects, while message passing only sees the co-occurrence graph.

Future Works

- Extend the graph to a much larger cap, e.g. 500 $\rightarrow$ 10,000 nodes.
- Vary the forecast horizon Δ, add multi-relational graphs aligned with the target (e.g. citation graphs).
- Optimize hyperparameters per model family (about 1.3% improvement for `graph_base` a posteriori).
- Better clarify why a simple drift model performs so well.
- Revisit loss functions and metrics (RMSE too punitive on a few large misses and probably less robust to distribution shift accross train/val/test).

Analyst-Facing Inference Ranking (Top 20)

Raw-view (NoGraph infer, October 2027)

Rank	Keyword	Score
1	leo satellite	27.818
2	leo	27.408
3	low earth orbit	27.091
4	gnss	19.438
5	navigation satellite	16.426
6	global navigation	15.405
7	real time	10.161
8	satellite networks	9.952
9	satellite communication	7.215
10	navigation satellite systems	6.545
11	terrestrial networks	6.228
12	point positioning	5.724
13	leo satellite networks	5.590
14	satellite systems	5.474
15	multiple access	5.110
16	satellite constellations	5.023
17	resource allocation	4.803
18	positioning accuracy	4.756
19	satellite terrestrial	4.669
20	inter satellite	4.556

Relative-view (Graph_multi_mix infer, October 2027)

Rank	Keyword	Score
1	low earth orbit	0.735
2	navigation satellite	0.623
3	global navigation	0.617
4	leo satellite	0.523
5	real time	0.384
6	leo	0.373
7	navigation satellite systems	0.312
8	point positioning	0.306
9	positioning accuracy	0.293
10	precise point positioning	0.287
11	precise point	0.286
12	high precision	0.283
13	non terrestrial	0.265
14	point positioning ppp	0.254
15	root mean	0.248
16	leo satellite networks	0.247
17	non orthogonal multiple	0.246
18	inter satellite	0.243
19	orthogonal multiple access	0.242
20	root mean square	0.241