Semantic topologies in the recursive application of generative AI models

Ben Swift[1] and Sungyeon Hong[2]

***Abstract*— Text-to-image and image-to-text models allow automated (but imperfect) semantic translation across modalities. This paper presents results and preliminary analysis of an empirical study of recursive information processing in popular open-weight generative artificial intelligence (genAI) models such as FluxSchnell and BLIP-2. Through clustering and topological data analysis we show some of the ways that different genAI models and initial prompts give rise to different semantic embedding trajectories, and suggest some ways forward for understanding how semantic information is transmitted through these types of complex information-processing systems.**

I. INTRODUCTION

Central to information theory [1] is the idea that information can be measured, compressed, and transmitted. Advances in so-called generative artificial intelligence (genAI), particularly text→image and image→text models, enable the automated translation of *semantic* information across modalities [2]. The CLIP-for-image-generation paper (the basis for OpenAI's DALL-E), trained on millions of image-text caption pairs from the internet, showed how to "produce variations of an image that preserve both its *semantics* and style" [3] (emphasis added). Semantic embedding techniques are useful outside of the generative text→image domain as empirical measures of semantic similarity between different texts [4].

Yet genAI models are imperfect semantic translators: they sometimes "fail" at their semantic translation task [5], as anyone who has used them can attest. The chance of such failures only increases with repeated transformations, like the children's game of Telephone. This paper attempts to trace how semantic information is preserved, altered, or lost across repeated genAI transformations. Using an empirical simulation approach, with genAI text/image models interconnected in different closed-loop feedback configurations, we generate text-image-text-$\cdots$ trajectories and study the semantics of these trajectories through text embeddings of these outputs. Over time, semantic content drifts or collapses, yielding trajectories of outputs shaped by the interplay of model parameters, initial prompts and emergent system dynamics.

Such recursive systems raise foundational questions at the intersection of cybernetics, genAI, and complexity science. Though individual outputs may appear unpredictable [6], the system exhibits underlying structures that hint at order emerging from apparent chaos. This paradox—between randomness and pattern—is a hallmark of feedback-driven systems. It invites inquiry into the nature of these emergent behaviours: are they chaotic, periodic, or governed by attractors in a high-dimensional phase space? What role do specific individual components or initial conditions play, and what is an emergent property of the system as a whole?

[1]Ben Swift and [2]Sungyeon Hong are with School of Cybernetics, College of Systems and Society, The Australian National University ben.swift@anu.edu.au, sungyeon.hong@anu.edu.au

II. METHODS

We performed a simple controlled experiment: a full factorial simulation over

- **four** distinct text-image-text closed-loop *networks*: all 2×2 combinations of two different text→image genAI models (FluxSchnell [7] and SDXL Turbo [8]) and two different image→text genAI models (BLIP-2 [9] and Moondream 2 [10])
- **45** different *initial prompts*: simple phrases e.g. "a cat", "a dog" (the full list is shown in the y-axis labels in Fig. 2)
- **four** different initial *random seeds* (because the text→image models will give different image outputs for the same text depending on the random seed)
- each text output is embedded independently with **three** different text *embedding models*: Nomic [11], RoBERTa [12] and MPNet [13]

In each case we construct trajectories by feeding the initial prompt and random seed into the first (text→image) genAI model in the network, then iteratively generate new (alternating) texts and images by feeding each output back in as input to the next genAI model in the network. While this could run forever, we stop after 1000 iterations. The result of each simulation run is therefore a length-1000 sequence of alternating text and image representations—see Fig. 1 for some examples.

The final step is to map each sequence into an embedding space, resulting in a sequence of vectors $\in \mathbb{R}^{768}$ which represents the "semantic trajectory" for that particular simulation run. In particular, the embedding models transform each text caption such that semantically similar captions are positioned close together in the embedding space. Although some have suggested [14] there is a "universal geometry of embeddings", we use three different popular open-source embedding models (Nomic Text v2, RoBERTa and MPNet) and therefore construct three independent embedding trajectories for each simulation run in an attempt to control for any artefacts introduced by the embedding models themselves.

Although multi-modal (text+image) embedding models exist [15], their text vs image embeddings are not directly comparable (we attempted to do so with [16] but the images were embedded differently enough to the texts that their clusters were entirely disjoint) so we choose to use text-only embedding models and only embed the text outputs (i.e.,

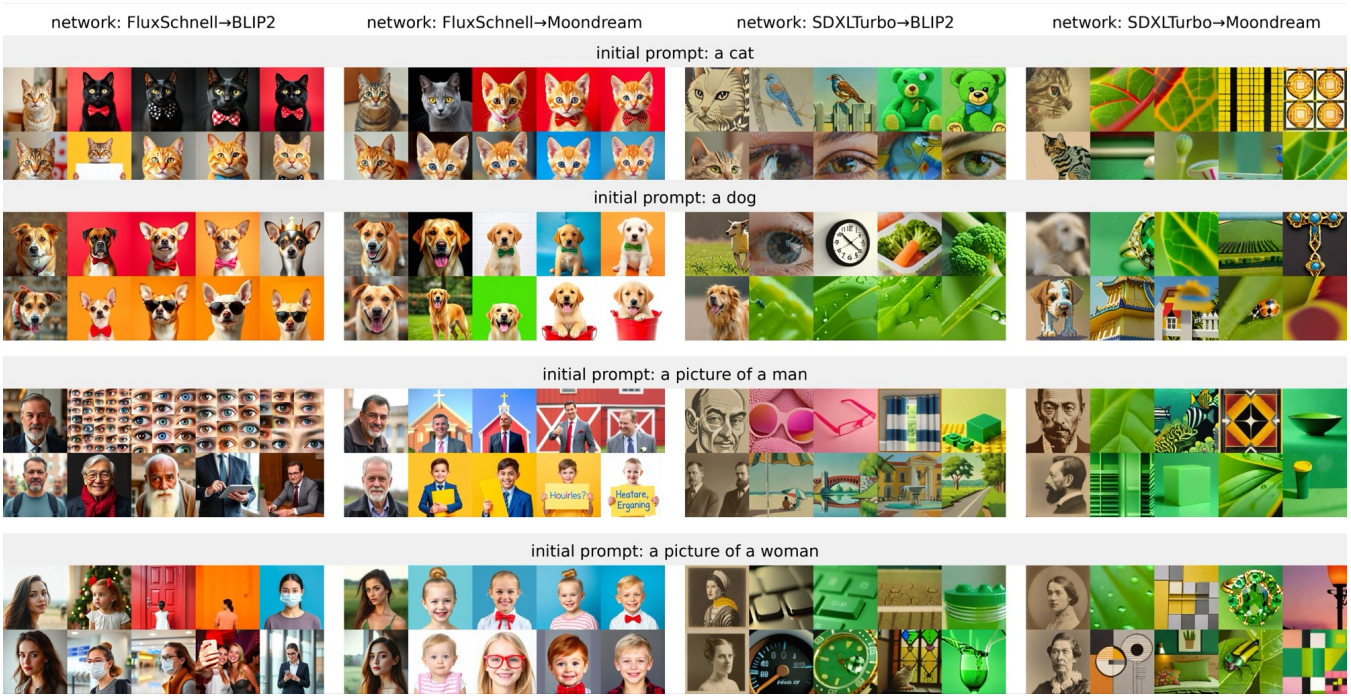

Fig. 1. Images from selected trajectories for different network + initial prompts. Within each block the left-hand image in each row is the first image in the trajectory (generated from the initial prompt), then the row contains every 250^{th} image until the right-hand image is the last image generated before the trajectory was terminated at 1000 iterations. There are obvious patterns—the left half of the figure (where FluxSchnell is the text→image model) are much more semantically stable, with the subject of the initial prompt in general still recognisable at the 1000^{th} iteration. The right half of the figure (where SDXL Turbo is the image→text model) shows trajectories that are less stable, with a preponderance of green, leafy imagery.

every second output of the alternating text-image-text-image sequence). Since we are interested in long-term behaviour, this is still sufficient to see the evolution of the semantic content from the initial prompt through the various genAI iterations.

In all cases we chose self-hosted, open-weight genAI models to enable batch-mode simulation, and preferred popular models (according to the HuggingFace Models Hub [17]). In parameterising the model pipelines we used default parameters wherever possible, and otherwise used values suggested by the model's official documentation.

These simulations were performed on commodity hardware with a single GPU (NVIDIA RTX 6000 ADA 48GB). All the code used to perform these simulations and analysis is available from `https://github.com/ANUcybernetics/panic-tda` (MIT Licensed). Time-lapse videos of the trajectories can be viewed at `https://www.youtube.com/playlist?list=PLvz7TESsjO5Raegv4pCTxPgWs5-pVJ6Er`.

III. RESULTS

Running these simulations resulted in 720 total trajectories (180 for each network) and 360k total embeddings for each embedding model. The computation was GPU-bound, took approximately 20 days of wall-clock time, and the data created (texts, images, embedding vectors and metadata) resulted in a ≈50GB SQLite database.

To analyse this trajectory data we used clustering (using Scikit-learn's `HDBSCAN` [18]) and persistent homology (using `giotto-ph` [19]). Each of these approaches considered the embedding vectors as a point cloud (partitioned by embedding model in the clustering case and by individual trajectory in the persistent homology case).

A. Clustering

The outlier rate was very high for all embedding models: Nomic 62.9%, MPNet 61.9% and RoBERTa 62.5%, indicating that the the embedding trajectories spend more than half the time in between the clusters. Across all trajectories, the top 5 most common clusters are shown in Table I. Note that the clustering process does not assign meaningful labels (just integers), so for labels we take the text content (i.e., the preimage of the embedding vector) of the cluster medoid. The top cluster label (containing 18.5% of all embeddings) does seem to describe many of the images shown in Fig. 1.

Fig. 2 shows how cluster membership differs between different initial prompts and networks. For space reasons, this figure shows only the Nomic embedding trajectories; the patterns for the other two embedding models appear qualitatively similar (although the clusters themselves are different).

To examine the "semantic stability" of a single embedding trajectory we consider the cluster run lengths—the number of consecutive embeddings which fall in the same cluster. Only the FluxSchnell→Moondream network exhibited stationary trajectories which stayed (for all 1000 iterations) in a single cluster—21 cases out of 180 total trajectories for this network. Of these, "goldfish" (7 trajectories) was the most

Fig. 2. Distribution of cluster membership for each embedding trajectory by initial prompt (y-axis) and cluster index (x-axis), faceted by network. Larger bubbles indicate that the embedding vectors for that network + prompt combination were more often found in that cluster (see Table I for index→text label mappings). The distinct vertical bands in the figure indicate that the embedding trajectories pass most commonly through a few "popular" clusters (especially cluster 1—the "green leaf. . ." cluster), and that these most popular clusters (and therefore the densest regions of the embedding space) are different for the four different networks.

Cluster label	Index	%
A close-up view of a vibrant green leaf features a diagonal line of veins and a small water droplet, set against a dark background.	1	18.5
A young girl with blonde hair and blue eyes smiles warmly, wearing a pink bow and a white shirt against a solid blue background.	17	5.3
A clownfish, with orange and white stripes, swims against a black background, displaying its distinctive fins and tail.	2	1.0
a yellow gold ring with an emerald stone	37	1.0
a painting of a colorful bird sitting on a branch	27	0.8

TABLE I

THE TOP 5 CLUSTER LABELS (NOMIC EMBEDDING)

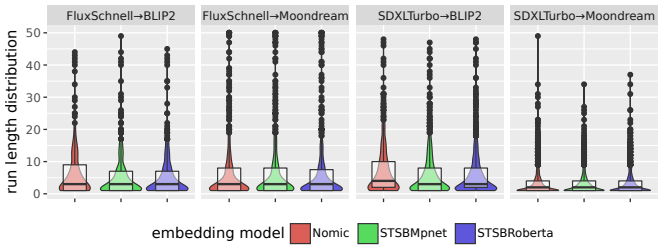

Fig. 3. This violin plot shows the distribution of cluster run lengths for different networks and embedding models. For all networks shorter run lengths dominate, although there are also clear outliers with very stable trajectories. The overall distribution seems similar across networks, with exception of the SDXL Turbo→Moondream network which seems to produce shorter run lengths.

common prompt. The distribution of cluster run lengths in shown in Fig. 3.

B. Persistent homology & persistence entropy

To effectively capture global patterns of the embedding trajectories beyond local clusters, we used persistent homology to capture topological features such as connected components (0-dimensional), loops (1-dimensional) and voids (2-dimensional) across a range of scales [20], [21]. Persistent homology is then computed to track the birth and death of topological features as a function of length scale, producing a persistence diagram for each trajectory. These diagrams offer a summary of the structural shape of the output dynamics and provide an interpretable way to detect recurring motifs, cyclic behaviours, or sudden phase transitions [22], [23].

To quantify the complexity of each trajectory's topological signature, we compute persistence entropy from its persistence diagram [24]. Persistence entropy measures the distributional uncertainty of topological features by treating each persistence interval (or the "lifetime" of a topological feature) as a contribution to an entropy score. High entropy reflects a diverse, transient topological structure, while low entropy indicates dominance of a few persistent features, suggesting possible attractor states or structural regularity.

The distribution of persistence entropy values across different networks and embedding models is shown in Fig. 4. Notably, the choice of embedding model does not substantially affect the outcome. The SDXL Turbo→Moondream network consistently exhibits higher entropy than the others. When considered alongside the images in Fig. 1, this

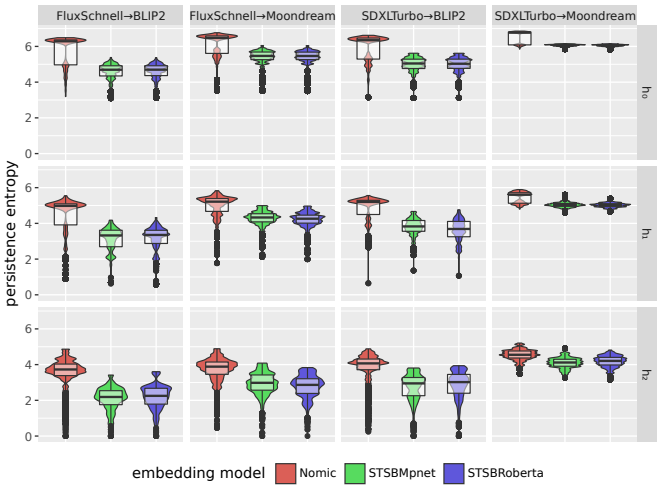

Fig. 4. This violin plot shows the distribution of persistence entropy for different networks and embedding models for topological features in dimension 0 (h_0) to 2 (h_2).

suggests greater semantic diversity and a broader expressive range throughout the iterative information transmission process within these trajectories. This observation is further supported by the cluster distributions in Fig. 2, where SDXL Turbo→Moondream shows a wider spread of cluster membership and reduced dominance of any single cluster—corresponding to a flatter entropy profile across homology dimensions in Fig. 4.

For Nomic embeddings, the persistence entropy distributions in Fig. 4 exhibit a clear bimodal pattern, which becomes less pronounced with increasing homology dimension. This suggests the existence of two distinct topological regimes in the system's dynamics. The low-entropy regime reflects the dominance of a few long-lived features, indicative of structured or attractor-like behaviour. In contrast, the high-entropy regime is characterised by a greater diversity of short-lived features, consistent with more exploratory or chaotic phases of information propagation. This bimodality may signal that the system transitions between ordered and disordered states during its iterative processing.

IV. DISCUSSION & CONCLUSIONS

These results suggest that there *is* structure in the embedding trajectories traced out by the repeated application of genAI models. There are many open questions: what is it about the specific genAI model architectures (or input prompts) which determines this structure? What other genAI models and prompts might yield similar (or different) insights under a similar simulation approach? How do we form testable hypotheses to study these phenomena from a classical (or otherwise) control perspective when exploring the joint state space of networked billion-parameter models? We hope that this work-in-progress report suggests some fruitful directions to pursue in tackling these questions.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[2] J. Conde, T. Cheung, G. Martínez, P. Reviriego, and R. Sarkar, "Analyzing recursiveness in multimodal generative artificial intelligence: Stability or divergence?" 2024. [Online]. Available: https://arxiv.org/abs/2409.16297

[3] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," 2022. [Online]. Available: https://arxiv.org/abs/2204.06125

[4] A. Deniz, M. Angin, and P. Angin, "Sentiment and context-refined word embeddings for sentiment analysis," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2021, pp. 927–932.

[5] M. A. and, "Making generative artificial intelligence a public problem. seeing publics and sociotechnical problem-making in three scenes of ai failure," *Javnost - The Public*, vol. 31, no. 1, pp. 89–105, 2024.

[6] A. Garliauskas, "Neural network chaos and computational algorithms of forecast in finance," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, vol. 2, Oct 1999, pp. 638–643 vol.2.

[7] B. F. Labs, "Flux," https://github.com/black-forest-labs/flux, 2024.

[8] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, "Adversarial diffusion distillation," in *European Conference on Computer Vision*. Springer, 2024, pp. 87–103.

[9] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," 2023. [Online]. Available: https://arxiv.org/abs/2301.12597

[10] V. Korrapati, "Moondream 2," GitHub repository, 2024. [Online]. Available: https://github.com/vikhyat/moondream

[11] Z. Nussbaum and B. Duderstadt, "Training sparse mixture of experts text embedding models," 2025. [Online]. Available: https://arxiv.org/abs/2502.07972

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[13] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," *arXiv preprint arXiv:2004.09297*, 2020.

[14] R. Jha, C. Zhang, V. Shmatikov, and J. X. Morris, "Harnessing the universal geometry of embeddings," 2025. [Online]. Available: https://arxiv.org/abs/2505.12540

[15] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "Imagebind: One embedding space to bind them all," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 15 180–15 190.

[16] Z. Nussbaum, B. Duderstadt, and A. Mulyar, "Nomic embed vision: Expanding the latent space," 2024. [Online]. Available: https://arxiv.org/abs/2406.18587

[17] Hugging Face, "Model hub," Web, 2024. [Online]. Available: https://huggingface.co/models

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] J. B. Pérez, S. Hauke, U. Lupo, M. Caorsi, and A. Dassatti, "giotto-ph: A python library for high-performance computation of persistent homology of vietoris–rips filtrations," 2021.

[20] H. Edelsbrunner and J. Harer, "Persistent homology—a survey," *Discrete & Computational Geometry - DCG*, vol. 453, 01 2008.

[21] F. Chazal and B. Michel, "An introduction to topological data analysis: Fundamental and practical aspects for data scientists," *Frontiers in Artificial Intelligence*, vol. Volume 4 - 2021, 2021.

[22] M. Lee and S. Lee, "Persistent homology analysis of ai-generated fractal patterns: A mathematical framework for evaluating geometric authenticity," *Fractal and Fractional*, vol. 8, no. 12, 2024.

[23] A. Myers, E. Munch, and F. A. Khasawneh, "Persistent homology of complex networks for dynamic state detection," *Phys. Rev. E*, vol. 100, p. 022314, Aug 2019.

[24] M. Rucco, R. Gonzalez-Diaz, M.-J. Jimenez, N. Atienza, C. Cristalli, E. Concettoni, A. Ferrante, and E. Merelli, "A new topological entropy-based approach for measuring similarities among piecewise linear functions," *Signal Process.*, vol. 134, no. C, p. 130–138, May 2017.