# Tracking the perspectives of interacting language models

**Hayden Helm**
Nomic AI
hayden@nomic.ai

**Brandon Duderstadt**
Nomic AI
brandon@nomic.ai

**Youngser Park**
Johns Hopkins University
youngser@jhu.edu

**Carey E. Priebe**
Johns Hopkins University
cep@jhu.edu

## Abstract

Large language models (LLMs) are capable of producing high quality information at unprecedented rates. As these models continue to entrench themselves in society, the content they produce will become increasingly pervasive in databases that are, in turn, incorporated into the pre-training data, fine-tuning data, retrieval data, etc. of other language models. In this paper we formalize the idea of a communication network of LLMs and introduce a method for representing the perspective of individual models within a collection of LLMs. Given these tools we systematically study information diffusion in the communication network of LLMs in various simulated settings.

## 1 Introduction

The success of large pre-trained models in natural language processing (Devlin et al., 2018), computer vision (Oquab et al., 2023), signal processing (Radford et al., 2023), among other domains (Jumper et al., 2021; Singer et al., 2022) across various computing and human benchmarks has brought them to the forefront of the technology-centric world. Given their ability to produce human-expert level responses for a large set of knowledge-based questions (Touvron et al., 2023; Achiam et al., 2023), the content they produce is often propagated throughout forums that have influence over other models and human users (Brinkmann et al., 2023). As such, it is important to develop sufficient frameworks and complementary tools to understand how information produced by these models affects the behavior of other models and human users. We refer to a system where a model can potentially influence other models as a system of interacting language models.

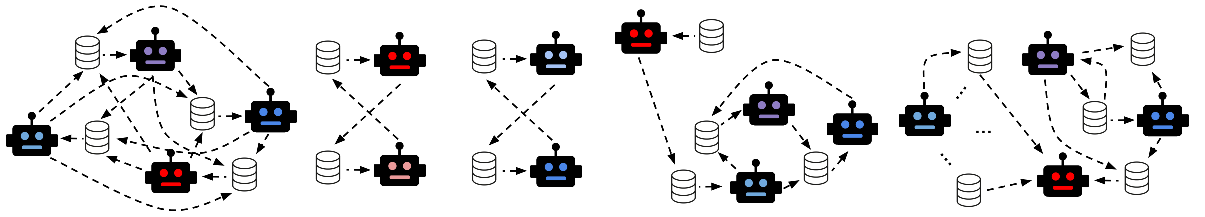
Beyond their ability to influence information on human-model forums, systems of interacting language models are interesting in their own right. Insofar as an individual model is an intriguing proxy for an individual human[1] (Helm et al., 2023), a system of interacting language models is an intriguing proxy for human communities. Systems of interacting language models are thus an alluring alternative or complement to studying human communities in the social sciences. For example, it is often infeasible or unethical to subject entire communities to different information paradigms to understand how individuals within the community – as well as the community itself – change in response to an intervention. These issues are less prominent for systems of interacting language models. Further, there is potential for greater control in community membership and cross-community interactions, which may improve reproducibility and mitigate the effects of sociological confounders.

In this paper, we study information diffusion in a system of interacting language models. We define information diffusion as the process by which information spreads and distorts across individuals or groups, typically through communication networks. The framework and methods that we develop can be applied to monitoring information diffusion in human-model forums and to the treatment of systems of interacting language models quantitatively as proxy human communities. The current standard (Perez et al., 2024) for studying information diffusion in a system of interacting language models requires i) parameterizing models with different system prompts, contexts, weights, or collections of data, ii) providing an environment or template for model-to-model or model-to-dataset interactions, and iii) analyzing how the outputs of the models change after a sequence of interactions.

For example, researchers include descriptions of desired model behavior or personality in the system prompt – e.g., “You have opinion $A$" is

[1] The content produced by natural language generative models is becoming indistinguishable from human-generated content.

*Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1508–1519
November 12-16, 2024 ©2024 Association for Computational Linguistics



(a) Fully connected. (b) Intra-class only. (c) Vulnerable. (d) General.

Figure 1: Examples of communication networks of language models and databases. The edge structure and model intitializations directly impact the evolution of the perspectives of the models and the overall health of the system.

included in the system prompt for model 1 and “You have opinion $B$" is included in the system prompt for model 2, etc. – to promote diversity in model response (Park et al., 2023; Chuang et al., 2023; Papachristou and Yuan, 2024). While the intended model response diversity is achieved, previous studies have failed to quantitatively assess the effect of different model initializations and, instead, rely on qualitative checks. Similarly, analyzing changes in model responses as the system evolves has previously been limited to human inspection of responses (Park et al., 2023), or classification of responses into a few classes (Chuang et al., 2023).

We introduce the *perspective space* of a collection of models to address the gap in quantitative methods for studying the diversity and evolution of model responses. The perspective space is an embedding-based representation of a collection of models designed to capture the relative differences in model responses for a fixed set of prompts. The method can be used to study information diffusion and general system dynamics by querying each model with the same set of queries at each time step. To demonstrate the effectiveness of the perspective space for understanding model-level diversity and for analyzing model-level and system dynamics, we formalize the system of interacting language models as a graph. The formalization enables systematic study of the effect of different communication structures on information diffusion that is otherwise not possible.

Our contribution is two-fold: i) We model a system of interacting language models as a graph and systematically study the effect of different communication structures on information diffusion. ii) We introduce the perspective space as a method to quantitatively analyze information diffusion in a population of language models.

## 2 A communication network of LLMs

Consider a system that consists of a collection of language models $\mathcal{F} = \{f_1, \dots, f_n\}$ and databases $\mathcal{D} = \{D_1, \dots, D_{n'}\}$. Given a set of prompts $\mathbf{X}$, systems deploying model $f \in \mathcal{F}$ may use the database $D \in \mathcal{D}$ – via fine-tuning, context retrieval, etc. – to produce more relevant outputs with respect to $\mathbf{X}$. The outputs of the updated model may be used to update a (potentially different) database $D' \in \mathcal{D}$. The updated database can then be used as a fine-tuning, retrieval, etc. database for a (potentially different) model $f' \in \mathcal{F}$. This set of interactions between a model and a database may occur across various models and various databases in the system.

As described, this system can be modeled as a graph $G = (V, E)$ where $V = \mathcal{F} \cup \mathcal{D}$ and the directed edge $(v, v')$ is in $E$ if vertex $v$ has influence on vertex $v'$. For example, the edge $(D, f)$ exists if $f$ has access to $D$ for retrieval augmentation or if it can use a subset of $D$ as fine-tuning data. Conversely, the edge $(f, D)$ exists if the output of $f$ can influence the content of dataset $D$.

Our primary interest is the dynamics of a system of interacting LLMs and databases where the vertex and edge sets are indexed by a discrete variable $t \in \{1, \dots, T\}$. There are many ways components of the graph may vary in $t$ in such a system. For example, the dataset $D^{(t)} \in V^{(t)}$ may be updated based on the outputs of the model $f^{(t)} \in V^{(t)}$ or the model $f^{(t)}$ may change after fine-tuning on the contents of the dataset $D^{(t)}$. In both cases $V^{(t)} \neq V^{(t+1)}$. Similarly, external factors such as the terms of use for a dataset may change to disallow its use for retrieval augmentation or a model may lose write-access to a dataset. In both cases $E^{(t)} \neq E^{(t+1)}$. Figure 1 illustrates simple examples of systems of LLMs as graphs, including three structures that are studied in the simulated settings in Section 4.

## 3 Defining a perspective space with surrogate data kernels

The system-of-LLMs-as-a-graph perspective provides a framework to systematically study the effect of different vertex sets and edge structures on the flow of information through the system as a function of $t$. The framework does not, however, provide a method to track the information flow. For this, we introduce an adaptation of the embedding-based data kernel presented in (Duderstadt et al., 2023). For our purposes, an embedding function $g$ is a mapping to real-valued vectors.

### 3.1 The data kernel & its surrogate

We let $\mathbf{X} = \{x_1, \ldots, x_m\}$ be a collection of prompts with $x \in \mathcal{X}$ and $f(\mathbf{X}) = \{f_\theta(x_1), \ldots, f(x_m)\}$ be the corresponding set of responses with $f(x) \in \mathcal{X}'$. Given an embedding function $g_i$ associated with $f_i$, the data kernel $A(g_i, \mathbf{X})$ of the evaluation dataset $\mathbf{X}$ captures the intrinsic geometry of the data with respect to $f_i$. The data kernel enables datum-level (i.e. comparing the representations of individual datums) and global level (i.e. comparing the holistic geometries of each model) comparisons of two models with potentially different architectures, sizes, etc. where direct comparison of $g_i(\mathbf{X}) = [g_i(x_1), \ldots, g_i(x_m)]^\top \in \mathbb{R}^{m\times p}$ and $g_j(\mathbf{X}) \in \mathbb{R}^{m\times p'}$ is otherwise not possible.

The methodology can be extended to compare the embedding spaces of multiple models $f_1, \ldots, f_n$ at once by considering the pairwise distance matrix of the corresponding data kernels. In particular, the classical multi-dimensional scaling (Torgerson, 1952)) of the $n \times n$ matrix M with entries $M_{ij} = ||\ A(g_i, \mathbf{X}) - A(g_j, \mathbf{X})\ ||_F$ yields $d$-dimensional Euclidean representations of the model $f_i$ with respect to $\mathbf{X}$. After this transformation, inference methods designed for Euclidean objects can be used for model-level analysis such as inferring differences in the training data mixtures.

The data kernel, as defined in (Duderstadt et al., 2023), requires the model $f_i$ to have an associated embedding function $g_i$. Unfortunately, for some state-of-the-art LLMs such as OpenAI's GPT series, Anthropic's Claude series, etc., an associated embedding function is unavailable and the data kernel cannot be constructed. To rectify this, we replace a model's associated embedding function with a *surrogate* embedding function $\tilde{g} : \mathcal{X}' \to \mathbb{R}^p$ that is not necessarily related to any of the LLMs under study.

The surrogate embedding function is not a drop-and-replace solution for model comparisons, however, since the embedding $\tilde{g}(\mathbf{X})$ is independent of $f_i$. Instead, we query the model with the elements of $\mathbf{X}$ and embed the responses $f_i(\mathbf{X})$ with $\tilde{g}$: the *surrogate data kernel* $A\left(\tilde{g}, f_i(\mathbf{X})\right)$ is simply $\tilde{g}\left(f_i(\mathbf{X})\right) \in \mathbb{R}^{m\times p}$. The surrogate data kernel is a $m \times p$ matrix representation of model $f_i$ with respect to $\tilde{g}$ and $\mathbf{X}$.

### 3.2 The perspective space

As with the original data kernel, we can use the surrogate data kernel to compare the responses from multiple models simultaneously via the CMDS of the pairwise distance matrix $\tilde{M}$ with entries $\tilde{M}_{ij} = ||\tilde{g}(f_i(\mathbf{X})) - \tilde{g}(f_j(\mathbf{X}))||_F$. We let $Z_i \in \mathbb{R}^d$ denote the $d$-dimensional vector representation of $f_i$.

Since the representations $Z_1, \ldots, Z_n$ are a function of the differences in the model responses – or “perspectives" – $f_1(\mathbf{X}), \ldots, f_n(\mathbf{X})$, we refer to the subspace populated by $\{Z_1, \ldots, Z_n\}$ as the *perspective space* of $\mathcal{F}$ with respect to $\mathbf{X}$. The information that is captured by the perspective space depends on $\tilde{g}$ and $\mathbf{X}$. In particular, $\tilde{g}$ needs to be able to distinguish between concepts that are intended to be distinguished. For example, a random mapping from $\mathcal{X}'$ to $\mathbb{R}^p$ is likely insufficient for comparing models, general-purpose embedding functions (Reimers and Gurevych, 2019; Nussbaum et al., 2024) should be sufficient for capturing the majority of signal, and domain-specific embedding functions (Risch and Krestel, 2019) should be used when the difference in models is highly nuanced. Similarly, $\mathbf{X}$ should contain prompts that the models are expected to have meaningfully different responses. We demonstrate this in Figure 2 where $\tilde{g}$ is fixed, $\mathcal{F}$ consists of 15 models (5 each from three different classes) and $\mathbf{X}$ is chosen to be relevant to the difference in classes (left) or “orthogonal" to the difference in classes (right). Models from the same class were fine-tuned on datasets with the same topic. The perspective space is more discriminative (i.e., the models from a given class cluster better) when $\mathbf{X}$ contains prompts relevant to the class-wise differences. More details related to the models shown in the two perspective spaces are provided in Appendix B.

The perspective space that includes the entire history of a system can be learned by considering the CMDS of the $|\mathcal{F}|^T \times |\mathcal{F}|^T$ pairwise distance ma-

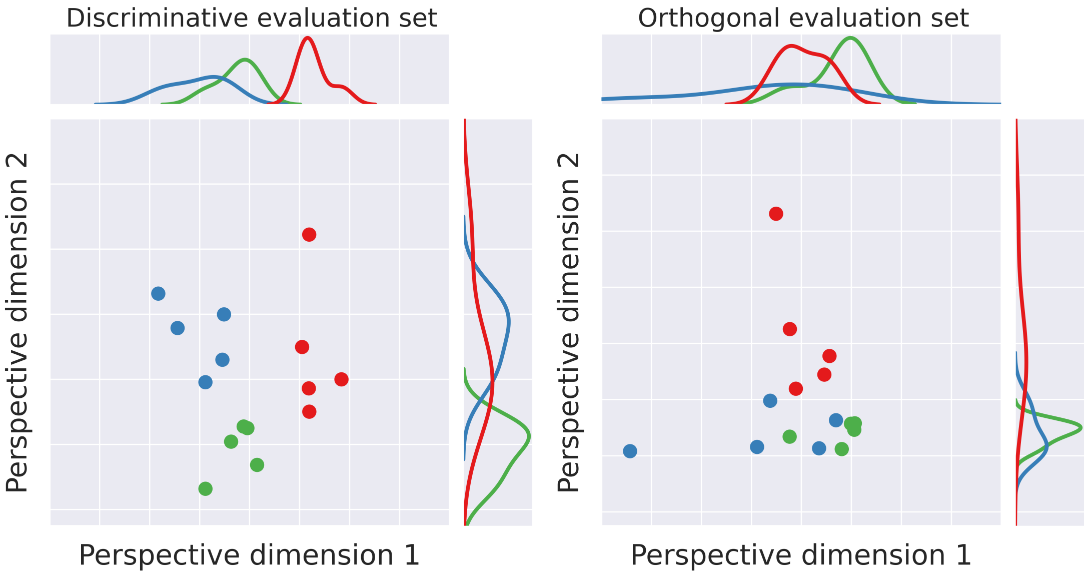


Figure 2: Two 2-d perspective spaces of fifteen models (5 models each from three classes, encoded by color). An evaluation set containing prompts relevant to the differences in the models (left) is better suited to induce a discriminative perspective space than an evaluation set containing "orthogonal" prompts.

trix with entries $||\tilde{g}(f_i^{(t)}(\mathbf{X})) - \tilde{g}(f_j^{(t')}(\mathbf{X}))||_F$ for all $i, j \in \{1, \ldots, |\mathcal{F}|\}$ and all $t, t' \in \{1, \ldots, T\}$. We use this perspective space when studying the systems below. The methodology can be extended to instances where only a partial history of the system is observed via out-of-sample methods (Bengio et al., 2003; Levin et al., 2018).

Throughout the next section we study the dynamics of a system of interacting language models through the lens of the first dimension of perspective space for visaulization purposes. We find that the dynamics of the first dimension correlates well with the change points in the system. In more complicated scenarios, it may be necessary to study perspective spaces with $d > 1$ to sufficiently capture system dynamics.

## 4 Simulating systems of interacting LLMs

We next simulate three different systems of interacting LLMs to demonstrate the effectiveness of the perspective space and its derivatives for capturing model and system dynamics for different underlying communication structures. The initial models in each system are based on an instance of the 410-million parameter model from the Pythia suite (Biderman et al., 2023) that has been instruction-tuned using Databricks' Dolly 15k (Conover et al., 2023). For each system we further fine-tune the base model on random question-pairs from setting specific topics from Yahoo! Answers (YA) dataset (Zhang et al., 2015) to promote response variation. We provide details on the instruction-tuning of the base model and the fine-tuning of the initial models in Appendix A and Appendix B, respectively. We use `all-MiniLM-L6-v2`, a sentence embedding function from (Reimers and Gurevych, 2019) based on (Wang et al., 2020b) hosted on the HuggingFace Hub (Wolf et al., 2020), as the surrogate embedding function and the implementation of CMDS from Graspologic (Chung et al., 2019).

In the three Case Studies (C.S.) we consider, each model interacts with another model in the system at each $t$. An interaction consists of model $i$ asking model $j \neq i$ a random set of questions from a fixed question bank and fine-tuning model $i$ using the resulting question-answer pairs as fine-tuning data. For a given $t$, the underlying communication structure $E^{(t)}$ determines which set of model interactions are possible for model $i$. In particular, the interviewed model $j$ is randomly selected from the set of models such that $(f_j, f_i) \in E^{(t)}$. The fixed question bank is used as the evaluation set to induce the perspective space.

While each system that we study technically consists of models and databases, each dataset is associated with only a single model. For convenience we discuss the systems as if the models themselves are directly connected. Our setting – where models are sequentially trained on each others outputs without intervention – can be viewed as a generalization of a single model sequentially trained on its

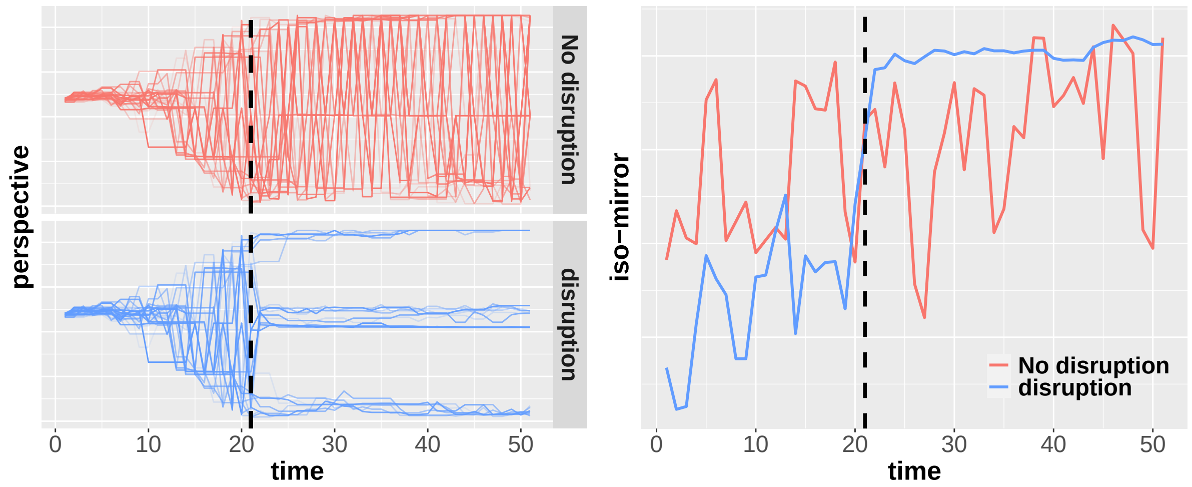


Figure 3: Tracking individual perspective (left) and system-level dynamics (right) of communication networks of chat-based language models with (bottom left) and without (top left) a disruption in communication structure.

own outputs as studied in (Shumailov et al., 2024).

At the end of each simulation setting we provide examples that motivated the case study.

### C.S. 1: Disrupting the communication network

We first study a system with $|\mathcal{F}| = 25$ models fine-tuned on different 400 random samples from YA with topic “Society & Culture" under two different system evolutions. For the first system evolution the underlying communication structure is unrestricted (i.e., $E^{(t)}$ fully connected, see Figure 1 “fully connected") for all $t$. For the second system evolution the underlying communication structure is unrestricted for $t < t^*$ and is then local-only (i.e., $(f_i, f_j) \in E^{(t)}$ only if model $i$ is model $j$'s nearest neighbor in perspective space after the interactions at $t - 1$) thereafter. We refer to the shift from unrestricted communication to local communication as a disruption in the communication structure.

At each time $t$ model $i$ asks 50 random questions from a question bank of 400 questions from YA with topic “Society & Culture". The initial 1-d perspectives of the models are relatively close to each other, as can be seen at $t = 0$ in both the top left and bottom left figures of Figure 3. As the system evolves for $t < t^*$, we observe the models “exploring" the perspective space. For the system that does not experience a disruption (top left), the exploration in perspective eventually stagnates and each model appears to oscillate between three different global perspective “sinks", one near the top of the figure, one in the middle of the figure, and one near the bottom of the figure. For the system that experiences a disruption at $t^* = 21$ (bottom left) the exploration in perspective space similarly stops, though the models do not oscillate between global sinks and, instead, persist in local sinks. The existence of multiple model sinks in both evolutions generalizes the behavior observed in (Shumailov et al., 2024), where the sequence of a single model sequentially trained on its own output converges to a single model sink in a process known as model collapse.

The difference in local and global sinks is quantified in Figure 4, where we report the number of clusters at each $t$ and the similarity of sequential cluster labels. We use Gaussian Mixture Modeling with the Bayesian Information Criterion (BIC) to estimate the number of clusters (Fraley and Raftery, 2002) and adjusted Rand index (ARI) to measure cluster label similarity. We find that the number of clusters for both systems eventually stabilizes and that the ARI between sequential cluster labels is lower for the global communication network after stabilization, which signifies higher cluster instability.

We quantify the evolution of the systems via the “iso-mirror" (Athreya et al., 2022), a system-level summary of the dynamics, in the right figure of Figure 3. The iso-mirror is an alternative to other summaries of system-level dynamics such as changes in the average perspective of all models that is better suited for systems where individual agent or subpopulation dynamics are non-uniform. In our setting, the iso-mirror corresponding to the system that does not experience a disruption is unstable throughout $t$. The iso-mirror corresponding to the disrupted system, however, clearly changes

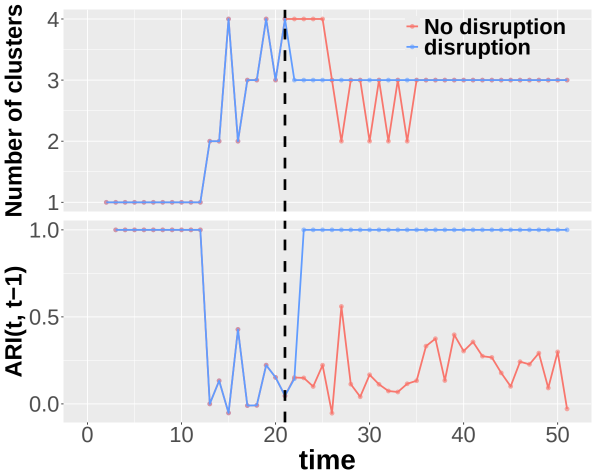


Figure 4: Estimated number of clusters found via GMM with BIC (top) and sequential ARI of cluster labels (bottom) for disrupted and undisrupted systems. The number of clusters in both systems stabilize, indicating the presence of model sinks. Model sinks are unstable in a system with no disruption and stable in a system with a disruption.

behavior at $t^*$ and remains constant throughout the remainder of its evolution.

**Motivating examples.** This case study was largely motivated by the COVID-19 pandemic (Zuzul et al., 2023) where social distancing, work from home, and social pods changed the latent communication structure for entire communities. It is also relevant to communication networks for range-limited devices where the definition of “local" depends on the geographical location of the device (Wang et al., 2020a).

## C.S. 2: Diffusion of an adversarial perspective

We next consider a system with $|\mathcal{F}| = 6$ models where five of the models are fine-tuned on a random set of 1000 question-answer pairs from YA with topic “Society & Culture" and the sixth is fine-tuned on a random set of 1000 question-answer pairs from YA with topic “Science & Mathematics". We refer to the model trained on data with topic “Science & Mathematics" as an “adversarial" model since it does not share the same initial perspective as the other five in expectation. A non-adversarial model is referred to as a “target" model at time $t$ if there is an edge from the adversarial model to it in $E^{(t)}$. Target models are randomly selected at the beginning of the evolution of the system and remain targets throughout a simulation. The evaluation set consists of 200 questions from the “Science & Mathematics" topic. At each iteration model $i$ asks model $j$ 100 questions.

For this experiment $E^{(t)}$ oscillates between two states. The first is a base state where the non-adversarial subnetwork is fully connected and there are no edges to or from the adversarial model. The second is a “vulnerable" state where there is an edge from the adversarial model to all target models, there are no other in-bound edges to the adversarial or target models, the non-target non-adversarial subnetwork is fully connected, and there are edges from the target models to the non-target models (see Figure 1 “vulnerable"). We simulate systems that have a vulnerable communication network once every two, five or ten iterations.

The trajectories of the 1-d perspectives of the models in the system with a vulnerable communication every other iteration are shown in the top of Figure 5 for systems with 0, 1, 2 and 5 targets. We also report the average perspective of the targeted models and the average perspective of the non-targeted models for each system.

For the system with no targets (top left) we observe similar behavior to the first case study under no disruption: the models initially explore the perspective space and eventually settle in a model sink. For the system with a single target we see the targeted model (top center left) oscillate between the adversarial perspective and the average perspective of the non-targeted models. Non-target models that interact with the target models immediately after the communication network was vulnerable are similarly pulled towards the adversarial perspective but to a lesser extent. Together these two effects limit the perspective exploration of the models in the system and eliminate the presence of the model sink.

For the system with two targets (top center right) the targeted models oscillate between the adversarial perspective and the average non-target perspective but the oscillations dampen as the non-target model perspectives start to drift towards the adversarial perspective. By $t = 20$ the average non-target perspective is closer to the adversarial perspective than its own starting position. That is, the entire system of LLMs has been compromised by the adversarial model targeting only a *minority* of the models in the system. The average perspective of models in a system with five targets (top right) quickly approaches the adversarial perspective.

In this setting we summarize system behavior via polarization defined as the difference in the aver-

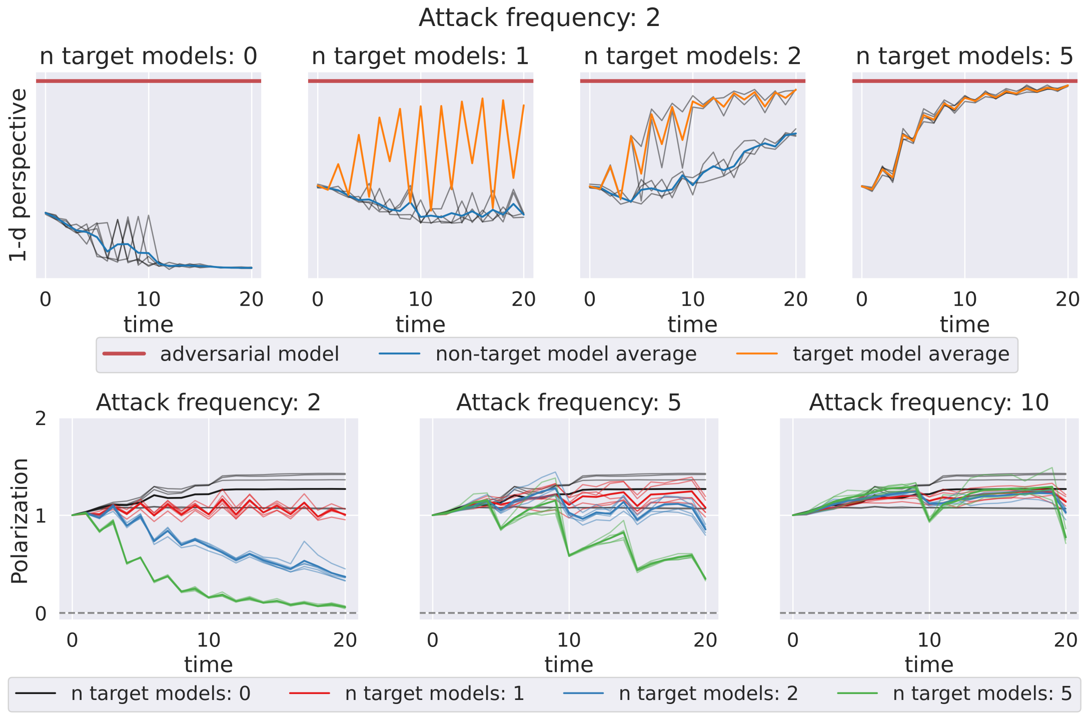


Figure 5: The evolution of 1-d perspectives of five interacting models where two models interact with an “adversarial" model every other interaction (top). Given enough nodes to influence, the adversarial model can compromise the entire network – as captured by the difference between the average 1-d perspective of the non-adversarial models and the 1-d perspective of the adversarial model for various amounts of target models and various attack frequencies (bottom).

age perspective of non-adversarial models and the perspective of the adversarial model normalized by this difference at $t = 0$. We report the polarization for five system initializations for vulnerable communication frequencies of two, five, and ten in the bottom of Figure 5, where for each initialization we consider a different set of 5 non-adversarial models. For example, for an attack frequency of two we see that polarization neatly summarizes our observations. In particular, the polarization increases when there are no target models, the polarization is relatively stable when there is a single target, the polarization slowly drifts towards zero when there are two targets, and the polarization quickly approaches zero when there are five targets. The system is more susceptible when more models are targeted for attack frequencies of five and ten, as well.

The trend across attack frequencies for a fixed number of target models indicates that given enough time between attacks the average model perspective is able to recover. This is likely due to the interaction mechanic involving a random subset of the evaluation questions – instead of the entire set – that enables system-level perspective homeostasis.

**Motivating example.** This case study was designed to mimic information diffusion in the presence of simple propaganda machines and to study how “attacks" on a minority affects the entire system.

### C.S. 3: Mitigating or promoting polarization

In our last case study we consider a system of $|\mathcal{F}| = 10$ models where five of the models are fine-tuned on 1000 random question-answer pairs from YA with topic “Society & Culture" and the other five are fine-tuned on 1000 random question-answer pairs from YA with topic “Science & Mathematics" . We let the topic in which the fine-tuning data is sampled from parameterize model “class". The evaluation set consists of 200 questions from each class. An interaction consists of model $i$ asking model $j$ 100 questions.

In this experiment we consider two different communication structures: unrestricted communication where $E^{(t)}$ is fully connected and intra-class only communication where $E^{(t)}$ consists of two unconnected class-wise fully connected subnetworks (see Figure 1 “intra-class only"). A system has the

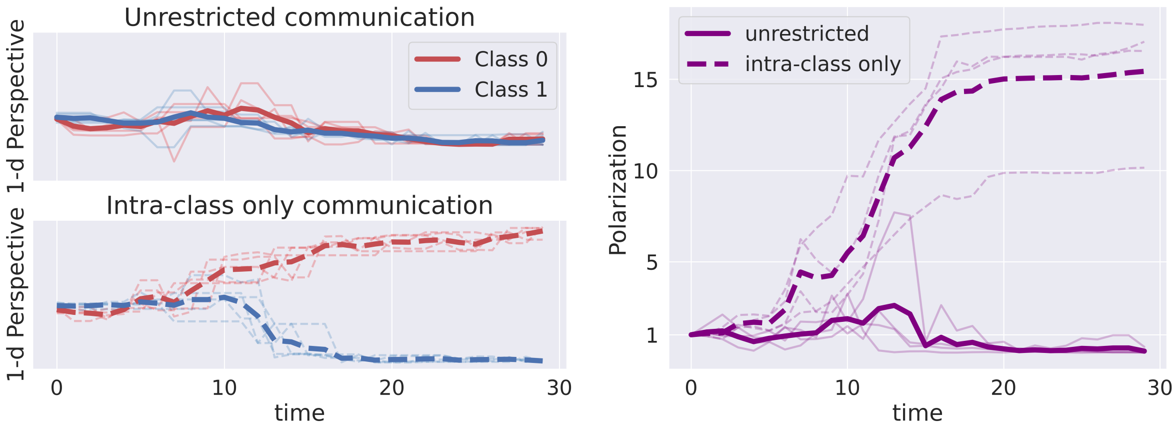


Figure 6: The evolution of 1-d perspective space representations of ten models from two classes under different underlying communication structures – unrestricted (left, top) and intra-class only (left, bottom). Class-wise average 1-d perspectives (bolded) are intertwined throughout the evolution of the system with unrestricted communication and diverge with intra-class only communication. Polarization captures this difference in behavior over multiple iterations of the experiment (right).

same communication structure for the entirety of its evolution. The top left figure of Figure 6 shows 1-d perspectives of the models in the system with unrestricted communication. Bolded lines represent the class average. As with fully connected communication network settings in the other case studies, we observe a period of perspective exploration before stabilizing. Notably, the two class-means stay intertwined throughout the entirety of the evolution of the system.

The bottom left figure of Figure 6 shows the evolution of 1-d perspectives with intra-class only communication. Under the intra-class only regime we see that the two classes explore *different* regions of the perspective space and eventually settle into two sinks with a much greater distance between them then the class-wise differences at $t = 0$. The polarization of the class-wise averages captures the distancing of the perspective “echo chambers", as reported in the right figure of Figure 6. Indeed, the polarization increased by 15x on average over four different simulation initializations under intra-class only communication. Average polarization is near zero by the end of the simulations under unrestricted communication.

**Motivating example.** This case study was designed to investigate the effect of “extreme" (e.g., intra-party communication only) underlying communication networks on two party systems.

## 5 Related Work

Our work is closely related to simulating groups of computational agents to study sociological and cultural phenomena (Steels, 1990; Wagner et al., 2003) and to continual learning (Vogelstein et al., 2020; Geisa et al., 2021). The former has seen renewed interest with the recent successes of LLMs. In particular, LLMs are – as of this writing – the computational tool that produces language artifacts most similar to ours and, as such, are an intriguing prospect for multi-agent sociological and cultural simulations. Recent work has included objective-less behavioral studies (Park et al., 2023), studying the formation of social networks (Papachristou and Yuan, 2024), tracking opinion dynamics via classification of LLM response (Chuang et al., 2023), and analyzing document collaboration (Perez et al., 2024). Our work extends these by introducing a framework to systematically study interventions and by introducing a quantitative method for tracking the evolution of agent perspectives.

Continual learning (Thrun, 1995, 1998) is largely concerned with how a single agent adapts to previously unseen inference tasks while avoiding “catastrophically forgetting" (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017) previous tasks. Our setting is slightly different, since we have multiple agents and no explicit task – though a large movement in perspective space is likely highly correlated to change in performance on language benchmarks related to the evaluation set. Indeed, large enough movements in perspective space and the emergence of model sinks when training a model recursively is related to catastrophic forgetting (Shumailov et al., 2024).

## 6 Conclusion

We introduced a system-of-LLMs-as-a-graph to enable systematic interventions to a system of interacting LLMs and the perspective space to quantitatively study the corresponding evolution of the system. We used these tools to highlight differences in paired systems across three case studies. For the particular interaction mechanic and update function that we used in our simulations, the model behaviors in perspective space consistently demonstrated initial model exploration and, in most cases, the emergence and persistence of model sinks. Further, we used derivatives of the perspective space such as the iso-mirror, polarization, and clustering to highlight differences in the evolution of paired systems.

For example, we observed differences in the iso-mirror (stable versus unstable after disruption) and clustering (global sinks versus local sinks after disruption) in the first case study; differences in the sensitivity of the average perspective of non-adversarial models to an adversarial perspective across number of victims and frequency of attack in the second case study; and differences in the behavior of polarization of two classes of models in the third case study.

## 7 Limitations

A system of interacting language models is a complicated system and, as such, analysis of them will often require simplification of aspects of the system. Our case studies are no expection. For example, the interaction mechanic (i.e., each model interacts with exactly one of its neighbors at time $t$) and update function (i.e., update model weights via fine-tuning) used in the simulations are more proof-of-concept than final-product in that they do not reflect our beliefs on how individuals within a community interact or “update" themselves, nor are currently deployed models constantly updated. While we do not attempt to enumerate all possible improvements here, we believe that it is imperative to work closely with social and cognitive scientists to understand the appropriateness of considering systems of LLMs as a proxy for human communities or online forums before generalizing observed simulated behavior to human-facing communities. Future work along these lines will include two major fronts: i) designing comprehensive statistical frameworks to understand the appropriateness of using a system of interacting LLMs as a proxy for various social settings and ii) extending simulation settings to include more sociologically plausible interaction and update mechanics.

Further, the simulation studies herein are but three system configurations worth considering. Indeed, of immediate interest is an extension to hierarchical social structures observed in large commercial and government institutions where the perspective space can be used to understand the effect of information injection, re-organizations, third-party seminars, etc. on individual-level, team-level, and organization-level dynamics.

There are also limitations related to the analysis in each of the three case studies we presented. For example, the first case study only investigated the difference between system behavior of global communication and global to hyper-local communication. More nuanced investigations into the effect of the number of models, the effect of the initializations of the models, the effect of the definition of “local", etc. are necessary to understand how the empirical observations may generalize to the real world. Similarly, for the second case study we only considered a single static adversarial model. A more realistic simulation might include multiple adversarial models, or adversarial models that change dynamically. For the third case study, if this analysis is to be used to understand polarization of political parties, it is necessary to understand the effect of cross-party communication, however rare it may be. We, again, believe that it is necessary to comprehensively explore each of these experiments before making claims about its applicability to society and human-model forums.

Lastly, we introduce the perspective space and demonstrate that it is sensitive to evaluation set. We do not, however, comprehensively explore or discuss potential applications or alternative model-based similarities. Similar methods have been used We expect the perspective space to be useful for various model-level inference tasks, as similar methods have been successfully used for classification (Chen et al., 2022) and change-point detection (Chen et al., 2023) in neuroscience applications. We also expect the model-based similarity most effective for capturing model differences will be system and task dependent (Eaton et al., 2008; Zamir et al., 2018; Helm et al., 2020).

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Avanti Athreya, Zachary Lubberts, Youngser Park, and Carey E Priebe. 2022. Discovering underlying dynamics in time series of networks. *arXiv preprint arXiv:2205.06877*.

Yoshua Bengio, Jean-françcois Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Roux, and Marie Ouimet. 2003. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Levin Brinkmann, Fabian Baumann, Jean-François Bonnefon, Maxime Derex, Thomas F. Müller, Anne-Marie Nussberger, Agnieszka Czaplicka, Alberto Acerbi, Thomas L. Griffiths, Joseph Henrich, Joel Z. Leibo, Richard McElreath, Pierre-Yves Oudeyer, Jonathan Stray, and Iyad Rahwan. 2023. Machine culture. *Nature Human Behaviour*, 7(11):1855–1868.

Guodong Chen, Hayden S Helm, Kate Lytvynets, Weiwei Yang, and Carey E Priebe. 2022. Mental state classification using multi-graph features. *Frontiers in Human Neuroscience*, 16:930291.

Tianyi Chen, Youngser Park, Ali Saad-Eldin, Zachary Lubberts, Avanti Athreya, Benjamin D Pedigo, Joshua T Vogelstein, Francesca Puppo, Gabriel A Silva, Alysson R Muotri, et al. 2023. Discovering a change point in a time series of organoid networks via the iso-mirror. *arXiv preprint arXiv:2303.04871*.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy T Rogers. 2023. Simulating opinion dynamics with networks of llm-based agents. *arXiv preprint arXiv:2311.09618*.

Jaewon Chung, Benjamin D Pedigo, Eric W Bridgeford, Bijan K Varjavand, Hayden S Helm, and Joshua T Vogelstein. 2019. Graspy: Graph statistics in python. *Journal of Machine Learning Research*, 20(158):1–7.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Brandon Duderstadt, Hayden S Helm, and Carey E Priebe. 2023. Comparing foundation models using data kernels. *arXiv preprint arXiv:2305.05126*.

Eric Eaton, Marie Desjardins, and Terran Lane. 2008. Modeling transfer relationships between learning tasks for improved inductive transfer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part I 19*, pages 317–332. Springer.

Chris Fraley and Adrian E Raftery. 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Ali Geisa, Ronak Mehta, Hayden S Helm, Jayanta Dey, Eric Eaton, Jeffery Dick, Carey E Priebe, and Joshua T Vogelstein. 2021. Towards a theory of out-of-distribution learning. *arXiv preprint arXiv:2109.14501*.

Hayden Helm, Carey E Priebe, and Weiwei Yang. 2023. A statistical turing test for generative models. *arXiv preprint arXiv:2309.08913*.

Hayden S Helm, Ronak D Mehta, Brandon Duderstadt, Weiwei Yang, Christoper M White, Ali Geisa, Joshua T Vogelstein, and Carey E Priebe. 2020. A partition-based similarity for classification distributions. *arXiv preprint arXiv:2011.06557*.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Keith Levin, Fred Roosta, Michael Mahoney, and Carey Priebe. 2018. Out-of-sample extension of graph adjacency spectral embedding. In *International Conference on Machine Learning*, pages 2975–2984. PMLR.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *Preprint*, arXiv:2402.01613.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Marios Papachristou and Yuan Yuan. 2024. Network formation and dynamics among multi-llms. *Preprint*, arXiv:2402.10659.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.

Jérémy Perez, Corentin Léger, Marcela Ovando-Tellez, Chris Foulon, Joan Dussauld, Pierre-Yves Oudeyer, and Clément Moulin-Frier. 2024. Cultural evolution in populations of large language models. *Preprint*, arXiv:2403.08882.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Julian Risch and Ralf Krestel. 2019. Domain-specific word embeddings for patent classification. *Data Technologies and Applications*, 53(1):108–122.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2024. The curse of recursion: Training on generated data makes models forget. *Preprint*, arXiv:2305.17493.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-a-video: Text-to-video generation without text-video data. *Preprint*, arXiv:2209.14792.

Luc Steels. 1990. Cooperation between distributed agents through self-orcamsation. In *Proceedings of the first European workshop on modelling autonomous agents in a multi-agent world*. Citeseer.

Sebastian Thrun. 1995. Is learning the n-th thing any easier than learning the first? *Advances in neural information processing systems*, 8.

Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer.

Warren S Torgerson. 1952. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Joshua T Vogelstein, Hayden S Helm, Ronak D Mehta, Jayanta Dey, Weiwei Yang, Bryan Tower, Will LeVine, Jonathan Larson, Chris White, and Carey E Priebe. 2020. A general approach to progressive learning. *Preprint at https://arxiv.org/abs/2004.12908*.

Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. 2003. Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1):37–69.

Fangxin Wang, Miao Zhang, Xiangxiang Wang, Xiaoqiang Ma, and Jiangchuan Liu. 2020a. Deep learning for edge computing applications: A state-of-the-art survey. *IEEE Access*, 8:58322–58336.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Tiona Zuzul, Emily Cox Pahnke, Jonathan Larson, Patrick Bourke, Nicholas Caurvina, Neha Parikh Shah, Fereshteh Amini, Jeffrey Weston, Youngser Park, Joshua Vogelstein, Christopher White, and Carey E. Priebe. 2023. Dynamic silos: Increased modularity in intra-organizational communication networks during the covid-19 pandemic. *Preprint*, arXiv:2104.00641.

# A Instruction-tuning Pythia-410m-deduped

The base model that we used in the case studies in Section 4 was an instruction-tuned version of the 410 million parameter model from the Pythia suite (Biderman et al., 2023). For instruction-tuning, we added three special tokens to its tokenizer's vocabulary, "### End", "### Instruction:", and "### Response:", and fine-tuned the model with a subset of Databricks' Dolly 15k (Conover et al., 2023). Each datum consists of an instruction, context, response, and category. We kept only data in the Open QA, Brainstorm, General QA, and Creative Writing categories and that had a response length less than 100 characters. This filtering left us with 1559 instruction-response pairs. We formatted a particular example as follows:

```
### Instruction: {instruction}
### Response: {response}
### End
```

We fine-tuned the model on the formatted data using Adam with a learning rate of $5 \times 10^{-5}$ and a batch size of 8 for 10 epochs. The final cross-entropy loss on the training data was $\approx 0.26$.

# B Case-study specific fine-tuning

For each of the case studies we further fine-tuned the instruction-tuned base model to promote response variation. For this, we used the data from the Yahoo! Answers (YA) dataset introduced in (Zhang et al., 2015), where each datum consists of a topic, a question title, question content, a list of answers, and a best answer. Given data from a particular topic, we further filtered the data by considering only examples with best answers less than 200 characters, with best answers that contained only a single sentence, and with question titles that contained only a single question. We formatted data from YA as follows:

```
### Instruction: {question title}
### Response: {best answer}
### End
```

Unless otherwise specified, fine-tuning is done using Adam with a learning rate of $5 \times 10^{-5}$. The initial models were trained for 3 epochs. The model updates after an interaction consisted of only a single epoch with a learning rate of $10^{-5}$.

To induce the perspective spaces shown in Figure 2 we trained 5 models each for three randomly selected topics. Each model was trained with 500 randomly selected examples.

## B.1 Case Study 1: Stochastically Equivalent Models

For case study 1, we randomly selected 400 examples with the topic "Society & Culture" that we used as both the evaluation set in the experiment and as a pool of data used for further sampling. In particular, we randomly sampled 200 samples from the set of 400 25 times and used the 25 subsets as fine-tuning data for different "stochastically equivalent" models.

## B.2 Case Studies 2 & 3: Two classes

For case studies 2 & 3, we considered filtered data from topics "Society & Culture" and "Science & Mathematics". For each topic we randomly sampled 1000 examples 10 times to use for fine-tuning.

For case study 2, we randomly selected a single model fine-tuned on "Science & Mathematics" to be the adversarial model. This model was the adversarial model for all system instances. We then randomly selected 5 models fine-tuned on "Society & Culture" data to be non-adversarial models. The non-adversarial models changed with each system instance.

For case study 3, we randomly selected 5 models from each class for every system instance.