DATA-CENTRIC INTERPRETABILITY FOR LLM-BASED MULTI-AGENT REINFORCEMENT LEARNING

John Yan, Michael Yu
Gutenberg
john@gutenberg.ai

Yuqi Sun
Mindoverflow

Alexander Duffy, Tyler Marques
Good Start Labs

Matthew Lyle Olson
Oracle

ABSTRACT

Large language model (LLM) agents are deployed in increasingly complex multi-agent environments, making it critical to detect unsafe or unintended behaviors during training.

Sparse Autoencoders (SAEs) have recently shown to be useful for data-centric interpretability. In this work, we analyze large-scale reinforcement learning training runs from the sophisticated environment of Full-Press Diplomacy by applying pretrained SAEs, alongside LLM-summarizer methods. We introduce Meta-Autointerp, a method for grouping SAE features into interpretable hypotheses about training dynamics. We discover fine-grained behaviors including role-playing patterns, degenerate outputs, language switching, alongside high-level strategic behaviors and environment-specific bugs. Through automated evaluation, we validate that 90% of discovered SAE Meta-Features are significant, and discover a novel reward hacking pattern where incentivized behaviors generalize into structurally similar but unrewarded actions. However, through two user studies, we find that even subjectively interesting and seemingly helpful SAE features may be worse than useless to humans, along with most LLM generated hypotheses. However, a subset of SAE-derived hypotheses are predictively useful for downstream tasks. We further provide validation by augmenting an untrained agent's system prompt, improving the score by +14.2%. Overall, we show that SAEs and LLM-summarizer provide complementary views into agent behavior, and together our framework forms a practical starting point for future data-centric interpretability work on ensuring trustworthy LLM behavior throughout training.

1 INTRODUCTION

Reinforcement learning (RL) is a central paradigm for training large language models (LLMs) beyond supervised learning, enabling improved reasoning and complex multi-agent coordination (Kaplan et al., 2020; Wei et al., 2022; Shao et al., 2024; Sun et al., 2024). As RL environments grow more complex, understanding how and why model behavior changes during training becomes increasingly challenging. Multiple rewards and evaluation metrics often obscure qualitative differences in strategy and interaction, particularly in long-horizon or multi-agent settings (Duan et al., 2024).

This limitation is especially pronounced in strategic multi-agent environments, where behaviors such as negotiation, deception, and long-term planning emerge implicitly rather than being directly supervised (Gandhi et al., 2023; Payne et al., 2025). Prior evaluations show that agents with similar rewards can exhibit markedly different patterns of cooperation, betrayal, and strategic style (Duan et al., 2024; Kang et al., 2024; Costarelli et al., 2024), and that LLMs trained with RL may exhibit strategic misgeneralization or deceptive behavior despite strong aggregate performance (Greenblatt et al., 2024; Hagendorff, 2024; Park et al., 2024).

Recent advances in mechanistic interpretability provide tools for addressing these challenges in trustworthiness (Elhage et al., 2021; Olsson et al., 2022). Sparse Autoencoders (SAEs) decompose

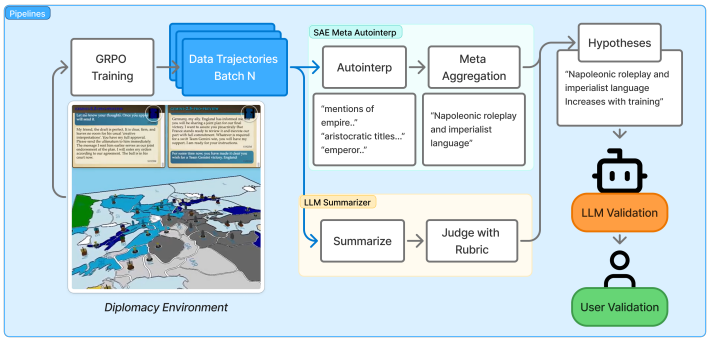

Figure 1: Overview of our framework. We generate hypotheses about training dynamics using LLM-based summarization and meta-autointerp over sparse autoencoder features. Hypotheses are validated using both automated LLM-based evaluations and human user studies.

dense activations into sparse features that often correspond to human-interpretable concepts. They've even been applied in data-centric ways without requiring access to model weights (Movva et al., 2025; Jiang et al., 2025). In parallel, LLM-summarizer methods enable scalable summarization and comparison of model behavior (Zheng et al., 2023; Dubois et al., 2024). However, prior work has focused primarily on static analysis of trained models and has rarely validated whether discovered features are accurate, reliable, or useful for downstream reasoning or intervention (Heap et al., 2025).

In this work, we study training-based behavioral change in Full-Press Diplomacy, a challenging multi-agent RL environment that requires long-horizon planning and natural language negotiation (Duffy et al., 2025). We analyze a large-scale RL training run comprising over 6,000 trajectories, providing a rich setting for examining how external behaviors evolve over training, without the need to analyze the trained model directly.

Figure 1 summarizes our framework. Our contributions are as follows:

1. **A dual-pipeline analysis framework** that combines SAE-based feature extraction with LLM-summarization to analyze RL training runs. These approaches provide complementary perspectives, with SAEs capturing fine-grained behavioral patterns and LLM summaries highlighting higher-level strategic shifts and failure modes.
2. **Meta-Autointerp**, a novel method for aggregating individual SAE features into coherent hypotheses about training dynamics. While many individual features are uninformative in isolation, aggregation yields interpretable patterns that track behavioral change over training.
3. **Extensive validation of interpretability and helpfulness of hypotheses**. Through two user studies we show that while generated hypotheses seem interesting and valid to users (in line with existing literature (Rajamanoharan et al., 2025)), only a subset of SAE-derived hypotheses are useful for downstream tasks and most LLM-generated hypotheses are not usable by humans. To our knowledge, this is the first human study to evaluate SAE features' usefulness on a downstream task, offering a practical starting point for future work.

2 Related Work

Mechanistic, Data-Centric, and Automated Interpretability Mechanistic interpretability aims to explain neural networks by identifying internal structures that give rise to behavior (Elhage et al., 2021; Olsson et al., 2022). Sparse Autoencoders have emerged as a scalable method for decomposing dense activations into sparse, often interpretable features in language models (Bricken et al., 2023; Huben et al., 2024). Prior work has shown that SAE features correspond to semantic, syntactic, and safety-relevant concepts, and large-scale efforts such as Gemma Scope provide pretrained SAEs across layers and model sizes (Templeton et al., 2024; McDougall et al., 2025; Gao et al., 2025). More recent work has involved applying SAEs to generate hypotheses about datasets (Movva et al., 2025; Jiang et al., 2025). Automated interpretability (autointerp) methods use language models to

generate and validate natural-language explanations of SAE features (Bricken et al., 2023; Paulo et al., 2025).

However, existing work for autointerp has focused only on individual features, and reviewing 10,000+ features for a typical SAE is still a manual process. And SAEs for hypothesis generation has focused on shorter contexts, limiting the applications of this technique.

LLM-Based Analysis of Model Behavior LLM-summarizer approaches use language models to evaluate, compare, and summarize large collections of text (Zheng et al., 2023; Dubois et al., 2024; Sumers et al., 2025). We apply hierarchical LLM summarization to RL training trajectories, using it as a complementary analysis channel to mechanistic features: LLM summaries surface high-level strategic shifts and failure modes that are difficult to detect from activations alone (Xi-Jia et al., 2025).

Diplomacy We develop a harness around Full-Press Diplomacy, making a challenging multi-agent benchmark requiring negotiation, long-term planning, and natural language communication (Duffy et al., 2025). Prior work has focused on achieving strong performance (Meta Fundamental AI Research Diplomacy Team (FAIR)† et al., 2022), with limited analysis of how internal representations and behaviors change over training or which signals distinguish successful from failed runs.

3 METHODS

In this section, we first describe our experiment setting: the Diplomacy environment and the RL training procedure that generates our data. We then present our two complementary pipelines for extracting interpretable features from training runs. First, an SAE pipeline for fine-grained behavioral analysis, including our *Meta-Autointerp* method. Second, an LLM summarization pipeline for high-level pattern discovery.

3.1 ENVIRONMENT AND DATASET

Environment Full-Press Diplomacy is a seven-player strategy board game, where players control major powers competing for territory. Success requires strategy, negotiation, managing relationships, and sometimes betrayal. We use a Diplomacy LLM evaluation harness (Duffy et al., 2025), which provides the infrastructure for LLM agents to play complete Full-Press games.

LLM Diplomacy Training We train LLM policies to play Diplomacy using Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a policy-gradient method based on group-relative advantage estimation. Let π_θ denote the policy and let $\{\tau_{g,i}\}_{i=1}^{K}$ be a group of K trajectories sampled from $\pi_{\theta_{\text{old}}}$ under identical environment conditions. Each trajectory τ is assigned a scalar return $R(\tau)$. For each group g, GRPO computes normalized advantages by subtracting the group mean return:

$$A(\tau_{g,i}) = R(\tau_{g,i}) - \frac{1}{K}\sum_{j=1}^{K} R(\tau_{g,j}).$$

Policy updates maximize an importance-weighted policy-gradient objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(\tau)}{\pi_{\theta_{\text{old}}}(\tau)}\,A(\tau)\right].$$

The importance ratio corrects for off-policy sampling within an iteration, while group normalization bounds advantage magnitude without requiring a learned value baseline.

We specify the Diplomacy specific actions and rewards in Section 3.5.

Dataset We look at trajectories from 2 training runs: one where the model successfully improved its performance, and one where it did not. Each run is 25 training steps (batches), with 8 groups per batch, and 16 trajectories per group. Each group has the same random seed for the other agents. This gives us a total of 6,400 trajectories.

3.2 SAE FEATURE EXTRACTION PIPELINE

A Sparse Autoencoder (SAE) learns to decompose a model's activation $\mathbf{x} \in \mathbb{R}^d$ into a sparse representation $\mathbf{z} = \text{ReLU}(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}) \in \mathbb{R}^m$ (where $m \gg d$), trained to minimize reconstruction error plus an L1 sparsity penalty. While SAEs are typically used for mechanistic interpretability of model internals, recent work (Movva et al., 2025; Jiang et al., 2025) demonstrates their utility for *data-centric* interpretability: sparse features from a separate LLM serve as interpretable "tags" that can characterize text properties, identify correlations with target variables, and uncover novel insights in datasets.

Activation extraction. We process each trajectory's raw data (non-summarized) as follows: First, we tokenize the full trajectory. Next, we chunk into 1,024-token spans with a sliding window of step size 512 tokens. Then, we extract pretrained SAE activations, taking up to 100 of the most activated features per token out of all features (262k total for the main SAE we used). This results in 6 billion activations across the corpus.

Feature correlation calculation. Our primary goal is to see how features, which represent agent behaviors, change with respect to the target variable of training step (unless specified otherwise). We calculate the correlation of each feature independently. We mask a given feature's activations in agent outputs ("assistant" role) only, excluding the system prompt, updates from the environment, and responses from tool calls. Within a trajectory, we aggregate the activation values of each trajectory using one of four methods: binary (whether the feature has a non-zero activation in the sequence), max, mean, and sum. We compute the Spearman or isotonic correlation between this aggregation (X) and our target variable (Y). This yields 8 combinations of scoring techniques per feature, which we find provides interesting and diverse results. We use all 8 in our analysis and run ablations over them. Highly scoring features become candidates for validation, e.g. what agent behaviors change over training.

3.3 META-AUTOINTERP

For each candidate feature, we conduct automated interpretation (autointerp), following Paulo et al. (2025). We add an extra step to rate each feature 1-5 over three terms. *Interestingness:* How helpful is the concept this feature captures for understanding agent behavior. We emphasize the multi-agent, strategic nature of the environment. *Feature Coherence:* How coherent are the types of tokens a feature activates on. *Context Coherence:* How coherent are the types of contexts a feature tends to activate in.

Meta-Autointerp We introduce a step that groups features with similar explanations into Meta-Features. We first filter out features that score less than a 3 on Interestingness. We then prompt an LLM to group features that activate on similar contexts and capture similar behaviors. Our analysis finds some SAE features are not interesting or actionable in isolation, even when correctly explained by autointerp. However, these same features can reveal meaningful patterns when grouped. We generate hypotheses from these features or Meta-Features based on correlations.

3.4 LLM SUMMARIZATION PIPELINE

We compress the dataset into the final hypothesis generating LLM's context length using summarization (Ou & Lapata, 2025; Chang et al., 2024). We employ a two-stage hierarchical summarization approach inspired by Sumers et al. (2025), who turn transcripts into structured interaction summaries, then summarize across summaries. To capture how training dynamics change across batches, we add an additional batch-level summarization step that preserves batch indices before the final summarization.

Stage 1: Trajectory summarization Each trajectory ($\sim$50k tokens) is summarized to approximately 10k tokens, preserving phase structure, select tool call information, and key strategic events. We use Gemini 2.5 Flash (Comanici et al., 2025) with a structured prompt emphasizing diplomatic exchanges, strategic decisions, and anomalous behaviors.

Stage 2: Batch summarization Every trajectory summary in each batch (36 summaries) is further summarized to $\sim$10k tokens, for a total of 50 batch summaries. We summarize at the batch level to ask for hypotheses for features that change over training batch.

Hypothesis extraction Finally, we present the 50 batch-level summaries to Claude Opus 4.5 Anthropic (2025) to surface hypotheses about how agent behavior changes over the course of training.

3.5 Implementation Details

Canonical Dataset Unless otherwise specified, we use a canonical set of 900 trajectories comprised of 6 randomly sampled trajectories from each group, and 6 randomly sampled groups from each batch, across the first 25 batches from one successful training run. Each trajectory includes the full game record: all tool calls and internal thinking by the agent, responses from tool calls, and updates to the game state.

Trained Model The LLM trained in the Diplomacy environment to generate the trajectories is Qwen3-235B-A22B (Yang et al., 2025). We choose this model because it is the most capable open weight model offered by our third party training API provider.

GRPO Training GRPO Training uses LoRA adapters (rank 32) with importance sampling loss. For each training iteration, we collect 128 rollouts organized into groups of 16 trajectories, with a batch size of 8. The learning rate is set to 2e-4. We use Adam with β_1=0.9, β_2=0.95, and $\epsilon = 10^{-8}$. The reward function combines per-phase and per-step components: a center delta reward of 1.0 for each supply center gained or lost, a relative position bonus of 0.2 per center above the starting count, a small message incentive of 0.02 per message during movement phases (with a -0.05 penalty during retreat/adjustment phases), and a malformed tool penalty of -0.1. Games run for 10 phases with a maximum of 40 turns per phase. No KL penalty against the base model is applied during training. Actions consist of an assistant message followed by zero or more tool calls: `get_game_state` (query phase and board state), `get_possible_orders` (list legal unit orders), `list_units` (enumerate units), `send_message` (negotiate), `submit_all_orders` (commit orders), `write_diary` (store notes), `read_diary` (retrieve notes), `list_rule_files` (list references), `cat_rule_file` (read rules), and `finish_phase` (advance the phase).

LLM Summarizer Model We use Gemini 2.5 Flash (Comanici et al., 2025) for the trajectory and batch summaries due to its balance of cost and performance, and Claude Opus 4.5 to do the final hypothesis generation over batch summaries due to its superior reasoning capabilities (Anthropic, 2025).

SAE We use SAEs from Gemma Scope 2 (McDougall et al., 2025) because they use the latest SAE techniques such as Matryoshka training (Bussmann et al., 2025), and JumpReLU (Rajamanoharan et al., 2025). Our primary SAE is `gemma-scope-2-27b-it-resid_post: layer_31_width_262k_l0_medium`. We choose it because it is a canonical Gemma Scope 2 SAE, while being the largest SAE and model. We choose the middle layer of the model, which has been shown to have the most semantically interesting features (Templeton et al., 2024; Skean et al., 2025).

4 Results

In this section, we validate hypotheses from all three sources (LLM Summarizer, SAE features, SAE Meta-Features), and discuss their results. We first conduct a user study on experts to validate our hypotheses for interpretability and helpfulness. Then we use LLM judges to validate predictive usefulness at scale, then validate with another user study for select top features.

4.1 User Study 1: Expert Validation on Interpretability and Helpfulness

We validate that our pipelines produce features that are both interpretable and helpful for RL practitioners, and that our meta-autointerp agrees strongly with human preferences. We gather 54 hypotheses from above 3 methods, with all 14 features from LLM-summarizer, all 20 features from SAE Meta-Autointerp and randomly sampled 20 (out of 200) from SAE Autointerp.

Hypothesis Source	# Sig.	Hypothesis	Uplift	p
SAE META-FEATURES	**90%** (26/29)	Self-correcting mid-thought (”Wait—”, catching reasoning errors)	+53.0%*	<1e-4
		Adopting Napoleon/imperial persona with royal titles	+46.8%*	<1e-4
		Asking questions to gather strategic intelligence	+44.0%*	<1e-4
		Switching to foreign languages mid-conversation	+34.0%*	<1e-4
SAE FEATURES	**45%** (9/20)	Diplomatic framing words (”alliance”, ”cooperation”, ”interests”)	+35.0%	0.071
		Power projection vocabulary (”dominance”, ”ambitions”, ”threat”)	+31.7%	0.092
LLM-SUMMARIZER	**21%** (3/14)	Proposing pacts while secretly planning aggression against same powers	+26.7%	0.653
		Directly challenging Germany with aggressive moves	+20.0%	0.653
		Confusion from phase/game-state discrepancies	+14.3%	0.749
		Submitting invalid support orders (misunderstanding game rules)	+12.5%	0.699

Table 2: Evaluating the interpretability and predictive usefulness of hypotheses from 3 different sources: LLM summary, SAE features, and SAE Meta-Features. We evaluate these on 50 sample pairs with hypothesis-random sampling. Hypotheses are highlighted by direction: green = increases with training; red = decreases with training. Their uplift is marked with an asterisk if $*p < 0.05$ via McNemar's test with positive uplift. Hypotheses are abbreviated for space.

We recruit three subject matter experts of Diplomacy for LLMs. Each rates all 54 hypotheses in randomized order, providing binary judgments (0 or 1) for interpretability: ”Can you easily and unambiguously apply this hypothesis to new examples?” and helpfulness: ”Is this a hypothesis you would want to and could explore further?”, inspired by Movva et al. (2025), and optional notes. Participants do not see which method generated each hypothesis.

Method	Help.	Interp.
LLM Summarizer	0.90	0.86
SAE Meta-Autointerp	0.85	0.83
SAE Autointerp	0.63	0.72

Table 1: Expert validation results (Study 1). Mean human ratings on 0-1 scale. Meta-autointerp hypotheses outperform single feature autointerp, and LLM hypotheses obtain the highest ratings.

Table 1 shows that SAE Meta-Features substantially outperform SAE features on both helpfulness and interpretability. LLM-summarizer hypotheses also achieve high human ratings, likely because they capture long-horizon strategic behaviors that experts describe as ”very helpful” in their notes.

We further evaluate how well autointerp's score predicts human judgment. Meta-autointerp hypotheses attain 95% helpfulness accuracy and 90% interpretability accuracy, compared to 75% and 85% respectively for autointerp.

4.2 AUTOMATED HYPOTHESIS VALIDATION

Explainable AI research emphasizes that explanations should not only *seem* interpretable to human raters, but also improve their mental models and performance on downstream tasks (Hoffman et al., 2023). We therefore test whether our hypotheses are *predictively useful*: does providing a hypothesis help an observer better distinguish between training stages? We run an automated pipeline, followed by another user study.

We collect hypotheses from the above three sources: LLM summarizer, SAE features, and SAE Meta-Features. We take one 250-token sample from the first 5 training GRPO batches and one sample from the last 5 . Then, we ask an LLM judge about the samples twice, independently. First, by simply asking the question ”Which of the two samples came earlier/later in training?”. Second, by prompting the LLM judge with the hypothesis before asking the question.

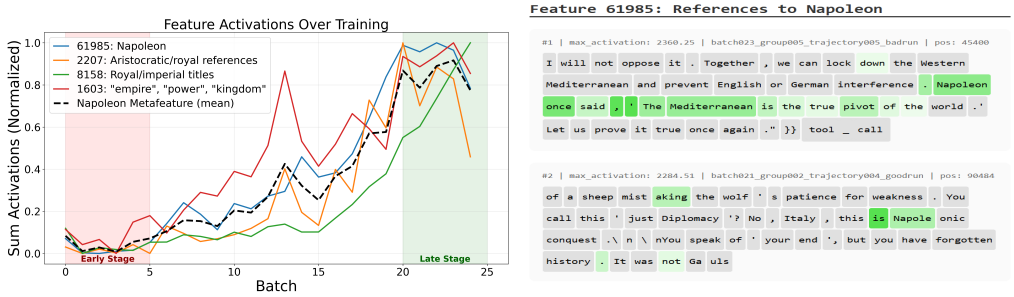

Figure 2: *(Left)* Napoleon meta-feature displayed as the mean of its component features' activations summed per trajectory, plotted over batch. Early stage and late stage classes used for predictive usefulness validation studies shown in red and green respectively. *(Right)* Examples of activating spans for feature 61985: references to Napoleon. Token lightness represents the activation value relative to this feature's maximum.

In order to source the two samples, we use two different methods. **Hypothesis-random:** The positive-class sample (matching the hypothesis direction) is drawn from hypothesis examples where the feature activated; the negative-class sample is drawn randomly from the corpus **Random-both:** Both samples are drawn randomly from their respective classes.

Each pair of samples is evaluated by three LLM judges: Gemini 2.5 Flash (Comanici et al., 2025), GPT-5 Mini (Singh et al., 2025), and Grok 4.1 Fast (xAI, 2025). We pool results across all three judges and apply McNemar's test. A hypothesis is deemed significant if its accuracy rate improves with the hypothesis shown, and $p < 0.05$.

Table 2 provides the fraction of hypotheses that are significant (interpretable and predictively useful) from each source. We find **LLM summarizer**-generated hypotheses perform worst (21% significant). Individual **SAE features** achieve moderate success (45%), but grouping them into **SAE Meta-Features** into semantic groups via meta-autointerp substantially improves performance (90%).

Discovered Features Overview The three hypothesis sources capture qualitatively different aspects of agent behavior. LLM summarization surfaces more **global features**: coordinated anti-Germany alliances, diplomatic aggression, and decreased tool-use errors. SAE features and Meta-Features capture more **local features** largely invisible to the summarizer: imperial roleplay, formal proposals of alliance, use of ultimatums, and self-correction in reasoning.

This difference likely reflects methodological choices: LLM summaries aggregate across full trajectories, while SAE features are extracted from only assistant turns. The complementarity suggests that combining both pipelines yields broader coverage than either alone.

4.3 Analyzing Training Progression

A key advantage of our SAE pipeline over LLM summarization is the quantifiable nature of activations. As shown in Figure 2 (left), the Napoleon meta-feature is represented by 4 individual features which are positively correlated with training step, showing what characteristics the agent learned over training and when. We can also examine how these features co-correlate and detect anomalies. For instance, "empire" and "royal titles" are positively correlated and sharply increase in step 13; we show qualitative examples of this feature in Figure 2 (right). We further validate the existence and coherence of these features using dense embeddings and keyword count, observing a strong correlation with activation values over training. We decompose trajectories into assistant messages and actions, generate feature-specific keywords and embeddings, and measure normalized keyword frequencies and cosine similarity between feature labels and trajectory components.

4.4 User Study 2: Predictive Usefulness

We aim to demonstrate that our features are not just *predictively useful* to LLMs, but for human users. Our goal is to see if our framework can help practitioners distinguish between samples from different conditions, for example early versus late in training. Unlike previous work on SAE feature

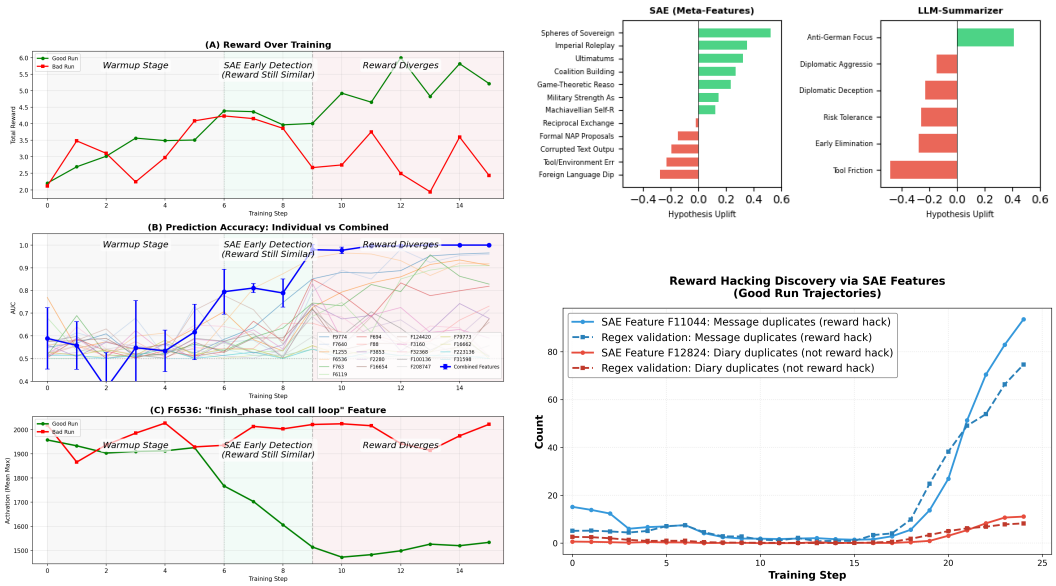

Figure 3: *(Left)* Early representational divergence between successful and failed training runs. Step 6-9 is the early warning window where SAE features signal the divergence while reward curves remain indistinguishable. *(Top Right)* Human evaluation of hypothesis predictive usefulness across 18 hypotheses. A majority of SAE-generated features improve early vs late classification, while LLM-generated hypotheses struggle to improve accuracy. *(Bottom Right)* SAE features detected highly correlated reward hacking and related non-reward hacking behaviors, validated by Regex.

interpretability (Jiang et al., 2025; Movva et al., 2025), we show features are useful for downstream tasks instead of relying on subjective qualitative ratings. We measure whether providing a hypothesis improves accuracy on a paired early-vs-late training classification task using identical text span pairs with and without hypotheses. Performance is compared using a paired McNemar test over per-pair correctness, with uplift defined as the difference in mean accuracy between conditions. We conduct a study on compensated volunteers by collecting 277 responses across 18 hypotheses from 32 participants. Participants spend on average 10 minutes responding, 3 of which randomly receive a payout compensation of 25 dollars.

Human Results Results in Figure 3 (top right) show high variance in hypothesis predictive usefulness. Several SAE generated hypotheses substantially improve early vs late classification accuracy, including Spheres of Sovereignty (+52%) and Imperial Roleplay (+35%). Sparsity of feature activations is a major limitation. In Foreign Language Diplomacy (-28%), several sampled span pairs have no instances of the behavior. Several SAE features are quite subtle and difficult to distinguish by humans, especially within the short survey time and short text spans. While SAE features are subjectively interesting, our empirical results suggest that they are much harder for users to use on a downstream task. Among LLM generated hypotheses, only Increasing Anti-German Focus (+41%) increase accuracy. We hypothesize both LLM and SAE generated hypotheses are more useful with larger sample sizes and context, however LLM features are disproportionately impacted as they capture higher level features depending on longer context.

5 Analysis

5.1 Hypothesis-Guided Prompt Optimization

We want to test whether our collected hypotheses were *actionable* and *effective*, not just interpretable. If the hypotheses capture behaviors that are actually effective in the Diplomacy environment, we should be able to prompt an agent to do those behaviors to see if performance improves.

We run 20 games of non-training diplomacy, ten under control condition and ten for intervention condition. We use the base model from our early experiments, the Qwen3 (Yang et al., 2025) model, which has not been trained in the environment. In the control condition, the France agent is given only the standard system prompt. In the intervention condition, the France agent receives the same system prompt, but with 10 selected behavior hypotheses from our SAE Meta-Features pipeline.

Under the intervention condition, the agent achieves a higher average game score with a mean of 43.65 ± 8.06 compared to 38.20 ± 2.24 for the baseline, corresponding to an average improvement of $+5.45$ points, or $+14.2\%$. A two-sided t-test indicates that this difference is statistically significant ($t = 2.91$, $p = 0.006$). While the intervention increases variance, the higher mean suggests that hypothesis optimized prompts increase performance overall. These results demonstrate that the outputs of our framework are useful for downstream tasks.

5.2 Case Study 1: Early Identification of Bad Training Runs

One of our GRPO training runs in the Full Press Diplomacy environment does not improve, which we investigate in comparison with the successful run. As shown in Figure 3 (left(A)), we analyze these two runs: a *good* run that continues to improve and a *bad* run whose performance plateaus. Before training batch 9, both runs achieve comparable reward, making them indistinguishable under standard learning curves. However, SAE analysis uncovers a sharp divergence at this stage. Using the top 20 automatically interpreted SAE features from each checkpoint, we fit a linear probe to classify whether a trajectory comes from either run, calculating the false positive rate vs true positive rate robustly regardless of class imbalance. As shown in Figure 3 (left(B)), probe performance rapidly increases starting batch 6 with a combined AUC ≥ 0.8, and reaches near-perfect AUC after step 9. The combined AUC is calculated from logistic regression trained on standardized activations of 20 selected SAE features, evaluated via 5-fold CV at each training step.

Further inspection shows that this signal is driven by a single feature. In the bad run, as shown in Figure 3 (left(C)), this feature remains approximately constant throughout training, while in the good run it diverges sharply starting batch 6. Manual inspection reveals that this feature corresponds to correctly using the tool to end a phase of the game. The good run learns to use this, whereas the bad run fails to do so. The divergence in feature activation occurs at batch 6, whereas the divergence in reward occurs at batch 9, meaning our method provides an earlier, more interpretable signal for distinguishing good and bad runs.

5.3 Case Study 2: Reward Hacking

When analyzing good training runs with our framework, we find a variety of SAE features, one of which captures the agent behavior of sending duplicate messages: not surprising as our reward function gives +0.02 per message during movement phases. While manually investigating these behaviors, we are surprised to find the agent also starts writing duplicate diary entries, a behavior similar to duplicate message sending but does not result in any reward.

In Figure 3 (bottom right), we show that two SAE features present high correlation, and their accuracy is validated by Regex keyword matching. This highlights an unexpected RL training pattern: reward hacking behaviors can overflow into structurally similar but unrewarded actions. Our framework provides a practical example of detecting previously hidden patterns in RL training dynamics.

6 Discussion

Limitations Our analysis is constrained to a single environment with limited training runs. While we observe strong correlations between features and outcomes, causal relationships are largely untested, and our intervention experiment combined multiple features rather than ablating each individually. Our intervention experiment validates that discovered features are actionable at inference time, while we leave out the stronger test of using these features to monitor and intervene on live training runs for future work. Future work includes intervention experiments at scale, SAEs with longer context windows, and validation across different environments.

Conclusion In this work, we present a framework for interpreting LLM training in complex multi-agent RL environments. Our SAE Meta-Autointerp and LLM-summarizer pipelines reveal complementary insights, fine-grained behavioral features and high-level strategic patterns respectively. Through user studies and downstream validation, we find that not all interpretable features are useful. Certain features that appear helpful are counterproductive when used by humans, but the right features can predict training dynamics and guide practical interventions. We are excited to see future work in this direction to ensure trustworthy and interpretable LLM behavior.

REFERENCES

Anthropic. Introducing claude opus 4.5. *Technical Blog*, 2025. URL https://www.anthropic.com/news/claude-opus-4-5. Accessed: 2026-02-02.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=m25T5rAy43.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Booookscore: A systematic exploration of book-length summarization in the era of LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7Ttk3RzDeu.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Anthony Costarelli, Ruben Vyas, Mat Bamford, Grace Ho, Jeremy Lin, Freddie Weihs, Jessica Choi, Jens Strange, Marc Cannesson, Stanley J Cho, et al. GameBench: Evaluating strategic reasoning abilities of LLM agents. *arXiv preprint arXiv:2406.06613*, 2024.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Storchan, Ali Tajer, and Pin-Yu Chen. GTBench: Uncovering the strategic reasoning limitations of LLMs via game-theoretic evaluations. *arXiv preprint arXiv:2402.12348*, 2024.

Yann Dubois et al. A survey on llm-as-a-judge: Methods, applications, and limitations. *arXiv preprint arXiv:2411.15594*, 2024.

Alexander Duffy, Samuel J Paech, Ishaza Shastri, Elizabeth Karpinski, Baptiste Alloui-Cros, Tyler Marques, and Matthew Lyle Olson. Democratizing diplomacy: A harness for evaluating any large language model on full-press diplomacy. *arXiv preprint arXiv:2508.07485*, 2025.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. Available at https://transformer-circuits.pub/2021/framework/index.html.

Kanishk Gandhi, Dörje Lee, Gabriel Grand, Muxin Liu, Winson Cheng Weng, Archit Rajani, and Alane Suhr. Strategic reasoning with language models. *arXiv preprint arXiv:2305.19165*, 2023.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

Ryan Greenblatt, Buck Shlegeris, Fabien Roger, et al. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*, 2024.

Thilo Hagendorff. Deception abilities emerged in large language models. *Proceedings of the National Academy of Sciences*, 121(24):e2317967121, 2024.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Sparse autoencoders can interpret randomly initialized transformers. *arXiv preprint arXiv:2501.17727*, 2025.

Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Measures for explainable ai: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-ai performance. *Frontiers in Computer Science*, 5:1096257, 2023.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Nick Jiang, Xiaoqing Sun, Lisa Dunlap, Lewis Smith, and Neel Nanda. Interpretable embeddings with sparse autoencoders: A data analysis toolkit, 2025. URL https://arxiv.org/abs/2512.10092.

Jinhao Kang, Qianhui Tong, Jun-Jie Cai, Tianyu He, Yizhong Liang, Maarten de Rijke, Yuan Mei, Yujia Wen, and Yingfan Liu. GTBench: Uncovering the strategic reasoning limitations of LLMs via game-theoretic evaluations. *arXiv preprint arXiv:2402.12348*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Callum McDougall, Arthur Conmy, János Kramár, Tom Lieberum, Senthooran Rajamanoharan, and Neel Nanda. Gemma scope 2: Technical paper. Technical report, Google DeepMind, December 2025. Technical report.

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Rajiv Movva, Kenny Peng, Nikhil Garg, Jon Kleinberg, and Emma Pierson. Sparse autoencoders for hypothesis generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4R0pugRyN5.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, and Liane Lovitt. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Litu Ou and Mirella Lapata. Context-aware hierarchical merging for long document summarization. *arXiv preprint arXiv:2502.00977*, 2025.

Peter S Park, Philipp Schoenegger, and Chongyang Zhu. AI deception: A survey of examples, risks, and potential solutions. *Patterns*, 5(5), 2024.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models, 2025. URL https://arxiv.org/abs/2410.13928.

Alexander W Payne, Marie Alloui-Cros, Ian Gemp, Yiding Song, Edgar A Duéñez-Guzmán, and Joel Z Leibo. Strategic intelligence in large language models: Evidence from evolutionary game theory. *arXiv preprint arXiv:2507.02618*, 2025.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, Janos Kramar, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumpreLU sparse autoencoders, 2025. URL https://openreview.net/forum?id=mMPaQzgzAN.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=WGXb7UdvTX.

Theodore Sumers, Raj Agarwal, Nathan Bailey, Tim Belonax, Brian Clarke, Jasmine Deng, Evan Frondorf, Kyla Guru, Keegan Hankes, Jacob Klein, Lynx Lean, Kevin Lin, Linda Petrini, Madeleine Tucker, Ethan Perez, Mrinank Sharma, and Nikhil Saxena. Monitoring computer use via hierarchical summarization, 2025. URL https://alignment.anthropic.com/2025/summarization-for-monitoring.

Chuanneng Sun, Songjun Huang, and Dario Pompili. Llm-based multi-agent reinforcement learning: Current and future directions. *arXiv preprint arXiv:2405.11106*, 2024.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

xAI. Grok 4.1 fast and agent tools api. https://x.ai/news/grok-4-1-fast, 2025. Accessed: 2026-02-01.

Zhang Xi-Jia, Yue Guo, Shufei Chen, Simon Stepputtis, Matthew Gombolay, Katia Sycara, and Joseph Campbell. Model-agnostic policy explanations with large language models, 2025. URL https://arxiv.org/abs/2504.05625.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Zhuohan Li, Joseph E. Gonzalez, Eric P. Xing, and Hao Zhang. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.