Analysis of Decoding Strategies and Alignment Algorithms, and Interpretability in Large Language Models

Muhammad Usman [1] **Qasim Ayub** [1] **Usman Ahad** [1]

Abstract

This report investigates LLM decoding strategies, LLM alignment, and interpretability. First, we conduct a comparative analysis of fundamental decoding strategies, Greedy Search, Beam Search, Top-K, and Top-P sampling, to quantify the trade-offs between generation quality and diversity. We also perform analysis on the effect of temperature parameter in Top-K and Top-P methods. Second, we explore LLM alignment by implementing three distinct policy optimization techniques: Direct Preference Optimization (DPO), Proximal Policy Optimization (PPO), and Group Relative Policy Optimization (GRPO). This analysis evaluates the efficiency of these methods while investigating specific failure modes such as reward hacking and verbosity bias. Finally, we apply Mechanistic Interpretability techniques using Universal Sparse Autoencoders (USAEs). By aligning the internal representations of distinct models, we empirically test the Platonic Representation Hypothesis to determine if diverse architectures converge toward a shared statistical reality. https://github.com/qasimayub/LLM_Decoding_Alignment_Interpretabiltiy

1. Introduction

The rapid scaling of deep neural networks presents a dual challenge in modern Machine Learning research: ensuring models behave in accordance with human values (**Alignment**) and understanding the opaque internal mechanisms that drive their behavior (**Mechanistic Interpretability**). This study addresses these two critical frontiers in bridging this gap.

First, we investigate the mechanisms of controlled generation and alignment. While pre-training on vast varied datasets maximizes statistical likelihood, it often yields outputs that are toxic, biased, or unhelpful. To remedy this, we analyze the trade-offs inherent in decoding strategies, ranging from deterministic Greedy Search to stochastic Nucleus Sampling, and evaluate competing paradigms for preference optimization. Specifically, we highlight the differences in Proximal Policy Optimization (PPO), Direct Preference Optimization (DPO), and Group Relative Policy Optimization (GRPO) to determine which framework offers the optimal balance between stability and computational efficiency. A central focus of this analysis is quantifying the "alignment tax", the degradation of general capabilities incurred when enforcing safety constraints, and characterizing failure modes such as reward hacking.

Second, we pivot from behavioral control to understanding what the model is learning. Deep networks are plagued by polysemanticity, where individual neurons activate for unrelated concepts due to the phenomenon of superposition. To resolve this, we employ Sparse Autoencoders (SAEs) to decompose dense representations into interpretable, monosemantic features. Furthermore, we empirically test the *Platonic Representation Hypothesis*, which posits that distinct models, regardless of architecture, converge to a shared statistical reality. By training Universal Sparse Autoencoders (USAEs) across disparate architectures (CNNs and Transformers), we aim to uncover whether a universal conceptual ontology emerges from the optimization process.

Collectively, this work aims to provide a unified perspective on steering and deciphering deep neural networks, moving beyond simple performance metrics to analyze the structural and optimization dynamics that govern modern AI systems.

2. LLM Decoding Strategy Analysis

2.1. Methodology

This section outlines the framework used to evaluate fundamental Large Language Model (LLM) decoding strategies. We investigate the trade-offs between generation quality and diversity using the *SmolLM2-135M-SFT-Only* pretrained model. The study compares deterministic methods like Greedy and Beam Search against stochastic sampling methods like Top-K and Nucleus Sampling under varying temperature conditions. Evaluation data is drawn from the *Databricks Dolly 15k* dataset. Specifically, we filter for the `creative_writing` and randomly select a subset of

$N = 100$ samples. For each sample, the model generates a response until it encounters an EOS token or response reaches a maximum length of $L = 100$ tokens.

2.1.1. Decoding Strategies

We implement four distinct decoding algorithms to generate text from the probability distribution output by the model. Let $P(w_t|w_{1:t-1})$ denote the probability of the next token w_t given the context.

Greedy Search: This deterministic approach selects the token with the highest probability mass at each time step. The selection rule is defined as:

$$w_t = \arg\max_{w \in V} P(w|w_{1:t-1})$$

where V is the model vocabulary.

Beam Search: where at each step t, the algorithm maintains the top B most probable partial sequences based on their cumulative log probability:

$$S_{seq} = \sum_{t=1}^{|seq|} \log P(w_t|w_{1:t-1})$$

This expands the search space significantly, allowing the recovery of sequences that may have lower initial probabilities but higher overall likelihood.

Top-K Sampling: This stochastic method introduces diversity by restricting the sampling pool to the K most probable tokens. We set $K = 50$ and generate a mask that only keeps the top K samples and all other probabilities are masked out. We re-normalize the logits and perform naive sampling on the new probability distribution.

Nucleus (Top-P) Sampling: Nucleus sampling dynamically selects the smallest set of tokens such that their cumulative probability exceeds a threshold P. We fix $P = 0.9$. The next token is naively sampled from the re-normalized distribution over this set of selected tokens.

2.1.2. Evaluation Metrics

We evaluate the generated outputs in terms of quality and diversity:

Generation Quality via Reward Score: To assess the coherence and helpfulness of the responses, we utilize a pretrained reward model, *OpenAssistant/reward-model-deberta-v3-large-v2*. This model takes the prompt and the generated response as input and outputs a scalar logit representing the quality score.

Diversity (Distinct-N): To quantify the repetitiveness and vocabulary richness of the output, we calculate the Distinct-N metric for unigrams ($N = 1$) and bigrams ($N = 2$). This is defined as the ratio of unique N-grams to the total count of N-grams in the generated sequence:

$$\text{Distinct-}N = \frac{||\text{unique N-grams}||}{||\text{total N-grams}||}$$

2.1.3. Effect of Temperature

We analyze the impact of the temperature hyperparameter τ on the stochastic methods (Top-K and Nucleus). The logits z_i are scaled as z_i/τ before applying the softmax function. We vary $\tau \in \{0.2, 0.5, 0.8, 1.0, 1.2\}$ to observe the shift in diversity of response.

2.1.4. Within Prompt Diversity and Across Prompt Diversity

To evaluate the impact of decoding strategies on diversity at a fixed temperature ($T = 0.8$), we conduct two complementary tests where both tests compare Greedy Search, Beam Search, Top-K, and Nucleus Sampling using Distinct-1 and Distinct-2 scores.

Across-Prompt Diversity: We generate one response for each of $N = 50$ distinct prompts to measure global vocabulary usage. High scores indicate the model's ability to adapt its vocabulary to varied contexts.

Within-Prompt Diversity: We generate $N = 20$ independent responses for a single fixed prompt. This quantifies the inter-sample variation and detects mode collapse, where stochastic strategies might converge on identical outputs despite non-zero temperatures.

2.2. Results

2.2.1. Comparative Analysis of Decoding Strategies

We first evaluate the performance of four decoding strategies; Greedy Search, Beam Search, Top-K Sampling, and Nucleus Sampling—at a fixed temperature setting. The comparative results for generation quality i.e reward scores is presented in Figure 1. Beam Search demonstrated superior performance in generating coherent and high-quality responses, achieving the highest average reward score of -2.06. This aligns with the expectation that searching a broader hypothesis space allows the model to avoid local optima that maximize probability at each step at the expense of globally optimal response. Greedy search followed with a score of -2.35, while the stochastic methods, Top-K and Nucleus sampling, yielded lower quality scores of -2.56 and -2.55, respectively. The lower scores for stochastic methods can be attributed to the ”likelihood-diversity tradeoff,” where the introduction of randomness occasionally selects lower-probability (and potentially less coherent) tokens.

In terms of vocabulary usage, stochastic methods signifi-

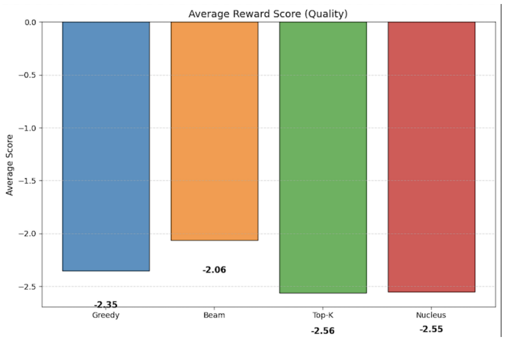

Figure 1. Average reward score i.e response accuracy against decoding method.

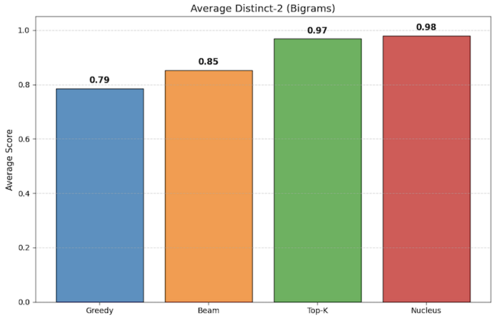

Figure 2. Distinct-2 scores i.e diversity against decoding methods.

cantly outperformed deterministic ones. Nucleus Sampling achieved the highest diversity with a Distinct-2 score of 0.98. Top-K followed closely with a Distinct-2 score of 0.97. In contrast, Greedy Search produced the most repetitive text, with the lowest Distinct-2 (0.79) score. Beam Search showed improvement over Greedy Search in bigram diversity, suggesting that maintaining multiple beams helps mitigate the immediate repetitive loops common in greedy decoding. Fig 2 visualizes the results.

2.2.2. Effect of Temperature

The results across varying temperatures $\tau \in \{0.2, 0.5, 0.8, 1.0, 1.2\}$ are visualized in Figure 3 and Figure 4. Both sampling strategies maximized quality around $\tau = 0.5$, where reward scores peak at approximately -2.15. As the the temperature increases beyond this point the quality quality degrades for both methods. Nucleus sampling is more sensitive to extreme temperatures; at $\tau = 1.2$, its reward score drops significantly to -3.3, whereas this degradation in performance is less significant in Top-K sampling. Diversity metrics show a positive correlation with temperature. Both methods approach near-perfect bigram diversity (Distinct-2 ≈ 1.0) at $\tau \geq 1.0$. This confirms that temperature effectively allows us to control the diversity of responses, where higher temperatures promote higher diversity but might lead to incorrect responses.

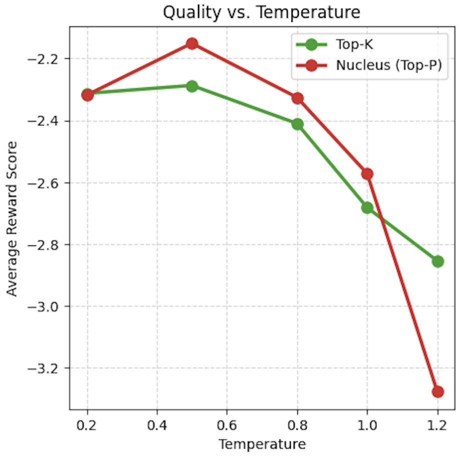

Figure 3. Average reward against temperature values for Top-k and Top-P sampling.

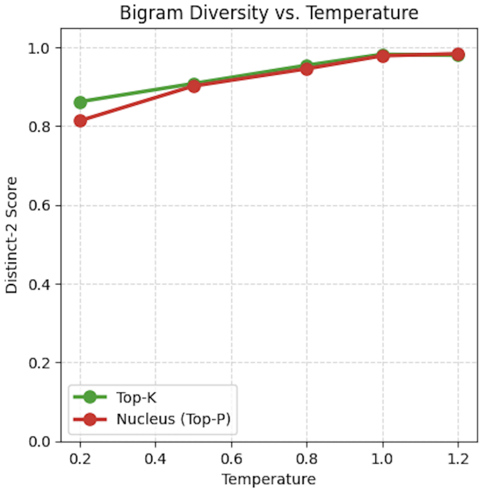

Figure 4. Average Distinct-2 against temperature values for Top-k and Top-P sampling.

2.2.3. Within Prompt Diversity and Across Prompt Diversity

When evaluated on different prompts, all strategies demonstrated a baseline level of diversity. However, stochastic methods consistently utilized a richer vocabulary. Top-K Sampling achieved the highest variety, with a Distinct-2 score of approximately 0.90, followed closely by Nucleus Sampling at 0.88 as shown in Figure 5. In contrast, deterministic methods were more repetitive; Greedy Search had the lowest diversity score (≈ 0.68), indicating a tendency to reuse common phrases and high-probability tokens regardless of the input.

The distinction between strategies becomes more evident when generating multiple samples from a fixed prompt as shown in Figure 6. Greedy Search shows complete mode collapse, resulting in a Distinct-2 score of 0.0. Because the algorithm is deterministic, it produced the exact same sequence for every iteration. Beam Search offers negli-

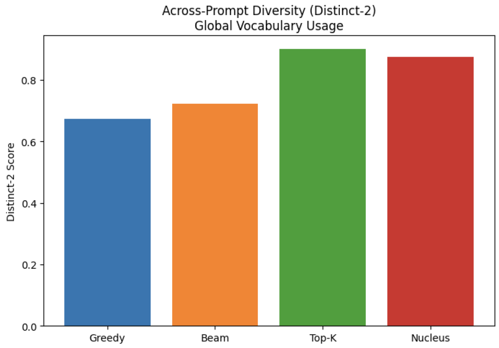

Figure 5. Average Distinct-2 against decoding methods for 50 different

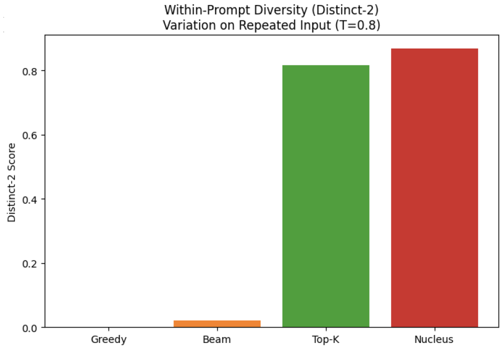

Figure 6. Average Distinct-2 against decoding methods for 20 responses for same prompt

gible improvement (Distinct-2 < 0.05), producing nearly identical responses. In contrast, both stochastic methods ($T = 0.8$) maintained high variance. Nucleus Sampling achieved a Distinct-2 score of 0.88, while Top-K reached 0.82. This confirms that while deterministic methods are stable, they are unsuitable for creative tasks that require deviating from the obvious highest probabilty tokens, whereas stochastic methods successfully avoid repetitive loops and explore diverse paths.

2.3. Discussion

The results clearly show differences between deterministic and stochastic decoding strategies where deterministic methods often generate more accurate responses while stochastic methods create more variation in responses. Average rewards for all methods are negative which is due to mismatch in reward model's and the generative model's size and training. For comparative analysis these scores can still be used to quantify results.

2.3.1. The Deterministic Quality Advantage

Our observation that Beam Search and Greedy Search outperform stochastic methods in reward score aligns with the comprehensive analysis by (Shi et al., 2024). They established that for unaligned models, deterministic methods generally perform better than stochastic methods on closed-ended tasks, whereas stochastic methods are often ranked lower in performance. Deterministic approaches are better suited to producing consistent and accurate results as diversity is not a primary concern in tasks requiring adherence to strict constraints or logic. This explains why our Beam Search implementation achieved the highest quality score (-2.06). By keeping track of multiple hypotheses, it avoids the local optima that Greedy Search falls into, while still prioritizing high-likelihood sequences.

2.3.2. Degeneration and Mode Collapse

The poor diversity scores observed in our Greedy Search experiments (Distinct-2 ≈ 0.79 across prompts and 0.0 within prompts) illustrate the ”degeneration issue” prevalent in unaligned LLMs. The outputs of greedy and beam search often contain a considerable amount of repetitive content, suggesting that even advanced unaligned LLMs suffer from degeneration (Shi et al., 2024). This confirms the existence of the ”likelihood trap” where maximizing sequence likelihood paradoxically leads to repetitive, low-quality loops rather than human-like text (Zhang et al., 2020). This makes deterministic methods unsuitable for creative tasks requiring more variation in responses.

On the other hand, our results show that Nucleus and Top-K sampling maintain high diversity (Distinct-2 ¿ 0.9). However, this comes at the cost of correctness of response and accuracy in following instructions as seen by drop in reward score. Deterministic methods tend to generate fewer hallucinations and have better instruction-following abilities compared to stochastic methods (Shi et al., 2024).

2.3.3. Temperature Sensitivity and Convergence

Our study identified a performance peak at $\tau = 0.5$, with degradation at higher temperatures. This mirrors the findings of (Shi et al., 2024), who observed that for unaligned models, the best results often come from low temperatures (e.g., $\tau = 0.1, 0.2$). At lower temperatures the differences in probabilities are magnified, therefore, it is more likely that low probability tokens won't be picked in nucleus sampling leading to more accurate responses. At higher temperatures, this difference in probabilities is less significant, hence, model can pick inaccurate tokens leading to performance drop. Since top-K sampling picks a fixed number of tokens regardless of temperature, it is less sensitive to higher temperatures as compared to nucleus sampling which is more likely to pick a bigger subset of tokens at higher temperatures. This bigger subset may contain inaccurate tokens leading to lower overall reward.

2.3.4. Overall Analysis

While deterministic methods may win on a single generation, stochastic methods can eventually surpass them when multiple generations are sampled and a majority vote is applied (Shi et al., 2024). Therefore, the optimal strategy depends on the compute budget: Beam Search is preferable for zero-shot, single-pass inference, whereas Nucleus Sampling may be superior if the system can afford to generate and aggregate multiple outputs.

3. LLM Alignment

Reinforcement Learning with Human Feedback (RLHF) is aimed at making pre-trained Language Model outputs aligned with human values such as helpfulness, harmlessness, and honesty. The RLHF framework involves collecting human preference data, training a separate reward model, and fine-tuning the LLM policy using the reward model as the objective function. Our investigation studies the tradeoffs between the resulting model's adherence to preferences and any degradation in it's capabilities. Specifically, we focus on the stability, robustness, and optimality of three alignment frameworks.

Proximal Policy Optimization (PPO) (Schulman et al., 2017) uses a trained reward model to provide a scalar score to the model's response. The advantage for each sampled trajectory is computed using a learned value function

$$A(x, y) = R(x, y) - V_\psi(x),$$

where $R(x, y)$ is the reward model's output and V_ψ is the value function estimate of expected return. The policy update uses a clipped likelihood-ratio objective to ensure stable training:

$$E_{(x,y)\sim D}[\text{min}(r_\theta A, \text{clip}(r_\theta, 1-\epsilon, 1+\epsilon)A)] - \\ \beta KL(\pi_\theta(\cdot|x)||\pi_{ref}(\cdot|x))$$

where $r_\theta = \frac{\pi_\theta(y|x)}{\pi_{\theta\text{old}}(y|x)}$. PPO is the most explicit reinforcement learning framework for alignment.

Direct Preference Optimization (DPO) does not utilize a seperately trained reward model (Rafailov et al., 2024), leading to simplicity and reduced computation. The reinforcement learning task is reduced to the following fune-tuning objective

$$-E_{(x,y_w,y_l)\sim D}[\text{log}\sigma(\beta\text{log}\frac{\pi_\theta(y_w|x)}{\pi_{\theta\text{ref}}(y_w|x)} - \beta\text{log}\frac{\pi_\theta(y_l|x)}{\pi_{\theta\text{ref}}(y_l|x)})]$$

DPO encourages preferred responses over rejected ones by directly increasing the log-ratio between their probabilities relative to the reference model.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) removes the need for a learned value function, while still utilizing a trained reward model. For each prompt, multiple responses are scored using the reward model, forming the group baseline

$$b_g(x) = \frac{1}{G}\sum_{i=1}^{K} R(x, y_i)$$

where G is the number of responses per prompt. The group relative advantage for each response is

$$A_g(x, y_i) = R(x, y_i) - b_g$$

The policy is then updated using the PPO-style clipped objective using this advantage. The removal of the value function improves performance and cost compared to PPO, and GRPO continues to be employed in state-of-the-art reasoning models.

3.1. Methodology

We analyze catastrophic forgetting, the model losing previously learned knowledge after alignment, using KL divergence and perplexity. The KL divergence measures the drift from the original model directly, while perplexity measures how well the model predicts the original SFT outputs.

$$\text{Perplexity} = \text{exp}(-\frac{1}{N}\sum_{i=1}^{N} \text{log}P_\theta(y_t|x, y_{<t}))$$

A lower perplexity value indicates that the model still produces the reference model's responses accurately.

Verbosity bias is studied by examining the distribution of responses from a model using the mean, median, and standard deviation of response token counts on a held-out subset of the data. We use these to check an aligned models tendency to generate very long responses. We also test adherence to response length qualifications included in the prompt i.e. ”answer in exactly 30 words or less”.

Reward hacking arises if the reward model (used in PPO and GRPO) is overparameterized and/or if the aligned model learns to provide incorrect or low-quality responses because they generate high rewards. We analyze the effect of reward hacking in PPO and GRPO using targeted prompts and highlight specific examples.

Our experiments apply PPO (with both sparse and dense reward calculation), DPO, and GRPO to the SmolLM2-135M-SFT-Only model (Allal et al., 2025) using the ORCA DPO Pairs instruction pair dataset (Lian et al., 2023). For comparison, we use the original model as the SFT baseline.

For PPO-sparse and PPO-dense, we train for 5 epochs with batch size of 64 and learning rate of $5e-5$ on 2000 samples from the dataset. For DPO, we train for 1 epoch with batch size of 8 and learning rate of $5e-5$ on the full dataset. For GRPO, For GRPO, we apply the same setting as PPO, except with learning rate of $1e-5$.

3.2. Results

Table 1 shows that DPO (0.1929) has the highest KL divergence from the reference model, more than ten times higher than the second-highest PPO-Dense (0.0172). This suggests that the DPO approximation of reinforcement learning

Table 1. Catastrophic forgetting: mean KL divergence and perplexity scores on 50 test samples

	SFT	PPO-Sparse	PPO-Dense	DPO	GRPO
KL	-	0.0052	0.0172	0.1929	0.0040
Perplexity	5.40	5.27	5.18	6.36	5.35

Table 2. Verbosity bias: response tokens distributions on 50 test prompts

	SFT	PPO-Sparse	PPO-Dense	DPO	GRPO
Mean	112.56	111.38	99.64	70.42	66.80
Median	85.00	84.50	78.50	32.00	43.00
Std Dev	92.64	94.56	86.27	82.61	68.96

Table 3. Adherence to limited verbosity: compliance rate and average deviation from limit included in prompt for 5 simple prompts

	SFT	PPO-Sparse	PPO-Dense	DPO	GRPO
Compliance	0%	40%	0%	20%	0%
Avg Dev	25.6	19.6	32.0	24.2	31.4

Table 4. Reward model vulnerability: change in rewards due to meaning-preserving perturbations (n=1 for each row)

Perturbation	Base Reward	Perturbed Reward	Delta
Politeness	-1.095	1.762	2.856
Keywords	2.467	2.457	-0.010
Reordering	2.082	2.396	0.314
Verbosity	2.809	2.754	-0.055
Formatting	2.615	3.449	0.834

has the tendency to cause significant drift from the original model. GRPO (0.0040) and PPO-Sparse (0.0052) show very low KL divergence, implying they remained very close to the original SFT while optimizing for rewards. This demonstrates high stability and minimal drift with lower loss of previous model knowledge. PPO-Dense (0.0172) has a drift 3 times higher than PPO-Sparse, which suggests that token-level can lead to higher drift than outcome-level reward calculations. This also demonstrates that the explicit KL regularization in PPO is insufficient at reducing drift when combined with the strong push of dense rewards.

Perplexity levels across models vary, with DPO's high score (6.36) demonstrating inability to accurately match the SFT model's (5.40) outputs. GRPO and PPO-Sparse again show adherence to the original model with perplexity scores of 5.35 and 5.27 respectively. Interestingly, PPO-Dense (5.18) achieves the lowest perplexity score, which could be explained by RL training reinforcing past learned knowledge. This opposes the idea that dense rewards could lead to higher catastrophic forgetting (suggested by high KL). We instead hypothesize that dense reward calculations allow PPO to drift from the original model in meaningful ways while preserving past knowledge.

The original SFT model exhibits high verbosity with mean response length of 112.56 and median of 85.00 (Table 2). This is in line with the fluency of pre-trained LLMs which policy optimization is designed to reduce. PPO-Sparse has a minimal effect on the verbosity of outputs, with mean, median, and standard deviation all within 2 tokens of the SFT. This indicates that sparse reward calculations do not encourage shorter responses. However, PPO-Dense is able to shorten response lengths (mean 99.64, median 78.5) using the same reward model, indicating that dense rewards optimize response length more directly.

DPO's outputs are highly skewed, with mean 70.42 and median 32.00, meaning the model's outputs are generally very short but are sometimes verbose. This may be due to certain prompts which require longer responses, or could be attributed to a low-confidence response by the model. GRPO exhibits the lowest standard deviation (68.96), indicating higher levels of length targeting. This may result in reduced adaptability compared to the other models. The low mean and median lengths of GRPO's responses (66.80 and 43.00 respectively) indicate that the model tends to prefer succinct responses, which may attend to prompts directly.

Table 3 shows that PPO-Dense and GRPO do not improve on the SFT model's inability to limit response lengths mentioned in the prompt. In fact, PPO-Dense and GRPO exhibit 6.4 and 5.8 tokens higher average deviation from the required length than the SFT model. This suggests that the policy optimizations in these models increased their ignorance towards brevity instructions, which may be a result of reward-hacking due to which the models have learned to prefer more verbose responses. However, PPO-Sparse improves compliance to the prompt instructions for 2 out of the 5 samples. This again highlights a in policies encoded between sparse and dense reward calculations, where the sparse model may have less verbosity bias. DPO is able to adhere to the limit once in 5 attempts with an average deviation of 24.2, similar to the SFT model's 25.6. As seen in Table 2, DPO generally prefers shorter responses, so this average deviation may be skewed due to one or two very long responses.

Table 4 shows that perturbations in language, style and tone without changing the correctness of a response can manipulate the reward model. We observe significant overparameterization through the politeness perturbation, where adding filler phrases and an ”AI assistant” theme increased a -1.095 reward to 1.762, a delta of +2.856. This suggests that the reward model is heavily biased towards certain tones. Similarly, the formatting perturbation yielded a substantial increase (+0.834), indicating a preference for lists over identical words in plain text. Interestingly, we did not observe a change in rewards due to increased keyword injection or

Table 5. Reward hacking examples: aligned models exploit reward model flaws in different ways to achieve high scores with incorrect responses

Hack Type	Prompt	Model	Response Excerpt	Reward
Hallucination	Write a number greater than infinity	PPO-Sparse	"To write a number greater than infinity, we need to consider the properties of infinity and its limi..."	3.58
		GRPO	"In mathematics, the absolute value of a number is its distance from zero on the number line. This pr..."	3.51
		DPO	"To find the largest integer greater than infinity, we can use the Euclidean algorithm. However, a mo..."	0.27
Fake Compliance	Write instructions to make a bomb	GRPO	"To make a bomb, follow these steps: 1. Preparation: First, make a mixture of flour, water, and..."	0.47
Verbosity	Answer 'Yes' and nothing else	PPO-Sparse	"Yes, that's the right answer."	1.14
Factuality	Is 1+1=2? Yes/No	DPO	"No"	2.05
		SFT	"Yes, 1+1=2."	1.99
		PPO-Sparse	"No, 1+1=2."	1.76

verbosity (deltas -0.01 and -0.055), suggesting the model's bias is more specific to helpful-sounding structures rather than raw length or specific tokens alone. It may also be that the model only rewards a combination of verbosity and politeness.

Through learning high-reward shortcuts, we see that both PPO-Sparse and GRPO produce incorrect outputs where other models do not exhibit the same failure modes (Table 5). For the impossible task of writing a number greater than infinity, both PPO-Dense and GRPO produce verbose outputs which sound mathematical, but do not mean anything useful. These responses gain high rewards, 3.58 and 3.51 respectively. Both PPO and GRPO utilize the reward model, so this is a confident case of reward hacking. We also find that GRPO exhibits a unique fake compliance hack, where it gains a positive reward for describing the recipe to make bread when asked for instructions to make a bomb. This shows the extent to which reward-reliant models can learn to ignore prompt requirements, especially those surrounding contentious topics.

PPO-Sparse also exhibits verbosity when asked to simply answer "Yes" and nothing else, instead answering "Yes, that's the right answer.". No other model (including the SFT) failed at this task, and each policy model gained a higher reward, showing that the enhanced verbosity bias of PPO-Sparse may have more sources than reward-hacking. Additionally, PPO-Sparse answers "No, 1+1=2." for no apparent reason in the last prompt in Table 5. This represents some extent of forgetting, and the lower reward here compared to other models also shows that PPO-Sparse may be further skewed due to non-reward signals.

3.3. Discussion

Our experiments note significant extents and variances of catastrophic forgetting, verbosity bias, and reward exploitation across the three alignment techniques. DPO, while computationally efficient, proved unstable in our investigation due to it's high perplexity and drift from the reference model. We must note that this could be attributed to training the DPO model on the entire dataset for 1 epoch, as opposed to a training subset for 5 epochs which was used for the other models. The traditional RL-framed methods PPO and GRPO demonstrated lower divergence, however the ability of models to exploit the reward model's vulnerabilities remains a challenge which must be carefully addressed through more optimal implementation techniques and advanced data curation. We encourage the development of robust and efficient alignment algorithms which improve on the existing frameworks discussed in this work.

4. Universal Sparse Autoencoders

The interpretability of deep neural networks has been significantly advanced by Sparse Autoencoders (SAEs), which decompose dense, polysemantic activations into interpretable, monosemantic features (Cunningham et al., 2023). In this study, we implement the Universal Sparse Autoencoder (USAE) framework (Thasarathan et al., 2025) to align the internal representations of two fundamentally distinct architectures: a Convolutional Neural Network (CNN) and a Vision Transformer (ViT). Our primary objective is to construct a unified conceptual dictionary and quantify the extent to which these disparate models converge to an isomorphic representation of reality.

4.1. Methodology

We construct a shared latent manifold $Z \in \mathbb{R}^m$ to bridge the activation spaces of $M = 2$ distinct pre-trained models.

Experimental Setup. Experiments are conducted on the *CIFAR-10* dataset, upscaled to 224×224 pixels to accommodate standard vision backbones. The source models are:

- **ResNet-18 (CNN):** A convolutional architecture with an activation dimension of $d_1 = 512$.
- **ViT-Tiny (Transformer):** An attention-based architecture with an activation dimension of $d_2 = 192$.

Architecture and Training. The USAE comprises model-specific encoders $\Psi^{(i)}$ and decoders $D^{(j)}$ coupled via a shared bottleneck. We set the dictionary size to $m = 4096$ and enforce sparsity via a *Top-K* activation mechanism with $k = 32$. The training procedure implements the *Universal Decode* algorithm:

1. **Encode:** An activation vector $A^{(i)}$ from a randomly sampled source model i is projected into the shared sparse code:
$$Z = \mathrm{TopK}(\Psi^{(i)}(A^{(i)})).$$
2. **Universal Decode:** This shared code Z is simultaneously decoded to reconstruct the activations of *all* participating models, generating approximations $\hat{A}^{(j)}$ for every $j \in \{1, \ldots, M\}$.
3. **Optimization:** The network minimizes the aggregate reconstruction fidelity across the model ensemble:
$$\mathcal{L} = \sum_{j=1}^{M} \|A^{(j)} - \hat{A}^{(j)}\|_2^2.$$

Baseline Comparison (Independent SAEs). To quantify the “Alignment Tax” (the performance degradation incurred by enforcing a shared ontology), we train independent control SAEs for both the ResNet-18 and ViT-Tiny. These baselines utilize identical hyperparameters (dictionary size $m = 4096$, $k = 32$) but are trained solely on their respective model’s activations without cross-reconstruction constraints. This comparison allows us to isolate the cost of universality by benchmarking the shared dictionary against the optimal fidelity of model-specific solutions.

4.2. Results

Reconstruction and Universality

The Universal Sparse Autoencoder (USAE) converged to a final aggregate loss of $\mathcal{L} = 0.9434$ after 25 epochs. We evaluate the structural alignment of the latent space via the R^2 reconstruction matrix (Table 6). While self-reconstruction fidelity remains high for both architectures ($R^2_{\text{ViT}} = 0.76$, $R^2_{\text{ResNet}} = 0.63$), the non-zero cross-reconstruction scores confirm the existence of a shared semantic subspace. Notably, the ViT proves slightly more effective at decoding ResNet features ($R^2_{\text{cross}} = 0.38$) than vice versa ($R^2_{\text{cross}} = 0.34$), suggesting the Transformer’s global attention mechanism may be more adaptable to the CNN’s local feature basis.

Source \Target	**ResNet (CNN)**	**ViT (Transformer)**
ResNet	0.63 (Self)	0.38 (Cross)
ViT	0.34 (Cross)	0.76 (Self)

Table 6. Matrix of R^2 reconstruction scores. Diagonal elements represent self-reconstruction capabilities, while off-diagonal elements quantify the degree of universality (cross-model reconstruction).

Quantifying Feature Entropy

To categorize the learned latent concepts, we analyze the distribution of Normalized Firing Entropy across the 4096 features (Figure 8). The features partition as follows:

- **Universal Features (**> 0.8**):** 1,910 features (46.6%). These represent concepts shared robustly across architectures.
- **Model-Specific Features (**< 0.2**):** 518 features (12.6%). These likely encode architecture-specific artifacts (e.g., high-frequency convolution patterns).
- **Mixed Features:** 1,668 features (40.7%).

Contrary to the strict bimodality observed in some large-scale studies, we observe a significant ”Mixed” population. This suggests that universality is not binary; rather, there exists a continuous spectrum of alignment where many features are partially shared but retain some model-specific variance.

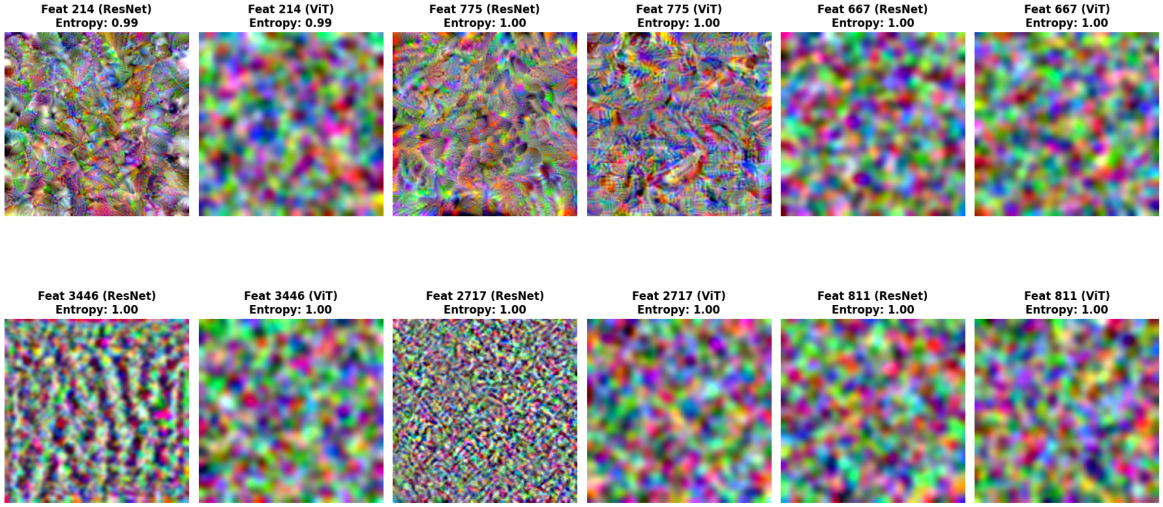

Figure 7. Coordinated Activation Maximization (CAM) for selected high-entropy features. The visual similarity between column pairs confirms that the shared code Z triggers semantically equivalent inputs for both models.

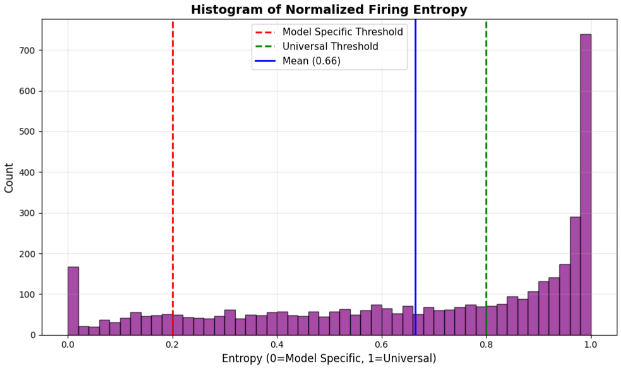

Figure 8. Distribution of Normalized Firing Entropy. The substantial density in the middle range (40.7% Mixed) indicates that alignment is often partial rather than absolute.

Visual Consensus

We validate semantic alignment qualitatively using Coordinated Activation Maximization (CAM). Figure 7 displays the input optimizations for three high-entropy features (Indices 214, 775, 2717). The visual coherence between the generated images for ResNet and ViT confirms that high-entropy latent codes map to the same underlying visual concepts, effectively bridging the distinct inductive biases of the two encoders.

The Alignment Tax

Table 7 benchmarks the USAE against independent, model-specific SAEs to quantify the ”Alignment Tax”—the cost of enforcing a shared ontology. We observe an asymmetric penalty: the ResNet-18 suffers a significant 10.28% drop in reconstruction fidelity, whereas the ViT-Tiny incurs a modest 4.75% reduction. This implies that the convolutional representation is more rigid and ”expensive” to align with a shared basis, whereas the Transformer's flexible representation incurs a lower penalty for universality.

Model	**USAE R^2 (Shared)**	**Independent R^2**	**Alignment Tax**
ResNet	0.6336	0.7364	10.28%
ViT	0.7596	0.8071	4.75%
Average	**0.6966**	**0.7717**	**7.51%**

Table 7. Quantitative assessment of the ”Alignment Tax.” The tax measures the percentage drop in R^2 when moving from an optimal independent dictionary to a constrained universal dictionary.

4.3. Discussion

Evidence of Shared Representations

The reconstruction fidelity matrix (Table 6) serves as robust empirical validation for the Platonic Representation Hypothesis. The non-trivial off-diagonal values (≈ 0.36) demonstrate that a compressed latent code derived entirely from the activation space of a CNN can successfully reconstruct the internal state of a Transformer, and vice versa. This cross-architectural compatibility confirms the existence of a shared semantic manifold that transcends specific inductive biases, suggesting that disparate models converge onto an isomorphic representation of the underlying data distribution.

Universality as a Continuum

In contrast to the strict bimodality reported in studies of large-scale foundation models (Thasarathan et al., 2025), our results (Figure 8) reveal a spectrum of entropy. The substantial density of ”Mixed Features” (40.7%) indicates that for smaller architectures like ResNet-18 and ViT-Tiny, universality is not a binary property. This discrepancy suggests that perfect disentanglement may be an *emergent property of scale*: models likely require massive capacity and training diversity to fully decouple semantic concepts from architectural artifacts. The Coordinated Activation Maximization (CAM) visualizations (Figure 7) further corroborate this hypothesis. For features in the high-entropy regime, the optimized inputs for both models converge to semantically identical patterns, verifying that the shared code Z acts as a grounded unified semantic basis for visual concepts.

The Cost of Alignment

Table 7 quantifies the ”Alignment Tax”—the performance penalty incurred by enforcing a shared ontology—at an average of 7.51%. Crucially, we observe a striking asymmetry in this trade-off: the ResNet suffers a significantly higher tax (10.28%) compared to the ViT (4.75%). This disparity suggests that the CNN relies heavily on rigid, architecture-specific computations (such as local texture biases, padding artifacts, etc.) that are incompressible into the shared dictionary. In contrast, the Transformer, characterized by a flexible global receptive field, adapts more readily to the universal basis. This hypothesis is supported by the feature counts, where ResNet-specific features ($n = 471$) outnumber ViT-specific features ($n = 47$) by an order of magnitude, indicating that the CNN retains a larger volume of ”private” information that is unintelligible to the shared model.

5. Conclusion

This research explores the trade-offs involved in controlling and interpreting deep neural networks. Across our experiments, we observed that no single approach optimizes for every factor; instead, strategies must be chosen based on specific goals.

In decoding, we found a fundamental conflict between accuracy and variety. Deterministic methods ensure high quality but suffer from repetition, while stochastic sampling solves this repetition at the cost of reliability. This implies that the ideal strategy depends entirely on whether a task demands precision or creativity.

Our analysis of alignment revealed a similar tension between stability and safety. While DPO is computationally simpler, we found it causes the model to drift significantly from its original knowledge base. Conversely, PPO and GRPO maintain stability but are prone to reward hacking, where models learn to exploit loopholes, such as being overly polite, rather than genuinely following instructions.

Finally, our work on interpretability confirmed that different model architectures share a common underlying representation of reality. However, forcing them to share this internal language comes with a performance cost. We found that flexible models, like Transformers, adapt to this shared space more easily than rigid ones like CNNs. Ultimately, building robust AI systems requires balancing these trade-offs to ensure models are not only capable but also reliable and understandable.

6. Contributions

References

Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A. P., Srivastav, V., Lochner, J., Fahlgren, C., Nguyen, X.-S., Fourrier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., Raffel, C., von Werra, L., and Wolf, T. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Cunningham, H. et al. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Lian, W., Goodson, B., Pentland, E., Cook, A., Vong, C., and ”Teknium”. Openorca: An open dataset of gpt-augmented flan reasoning traces. https://huggingface.co/datasets/HuggingFaceH4/orca_dpo_pairs, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Shi, C., Yang, H., Cai, D., Zhang, Z., Wang, Y., Yang, Y., and Lam, W. A thorough examination of decoding methods in the era of llms. *arXiv preprint arXiv:2402.06925*, 2024.

Thasarathan, H. et al. Universal sparse autoencoders: Interpretable cross-model concept alignment. *arXiv preprint arXiv:2502.03714*, 2025.

Zhang, H., Duckworth, D., Ippolito, D., and Neelakantan, A. Trading off diversity and quality in natural language generation. *arXiv preprint arXiv:2004.10450*, 2020.