Self-Improvement as Coherence Optimization: A Theoretical Account

Anonymous authors
Paper under double-blind review

Abstract

Can language models improve their accuracy without external supervision? Methods such as debate, bootstrap, and internal coherence maximization achieve this surprising feat, even matching golden finetuning performance. Yet why they work remains theoretically unclear. We show that they are all special cases of *coherence optimization*: finding a context-to-behavior mapping that's most compressible and jointly predictable. We prove that coherence optimization is equivalent to description-length regularization, and that among all such regularization schemes, it is optimal for semi-supervised learning when the regularizer is derived from a pretrained model. Our theory, supported by preliminary experiments, explains why feedback-free self-improvement works and predicts when it should succeed or fail.[1]

1 Introduction

Language models can improve their accuracy without external supervision. Methods such as debate (Irving et al., 2018; Khan et al., 2024), internal coherence maximization (ICM) (Wen et al., 2025),[2] iterative bootstrap (Lee et al., 2013; Xu et al., 2023), and Metropolis-Hastings sampling (Karan & Du, 2025) have matched supervised finetuning performance on ground-truth labels, despite using nothing but unsupervised questions and a pretrained model not finetuned on the present task. This is surprising. Where does the "supervision" come from? As we will show, the supervised signal is already embedded in the pretrained prior. The coherence optimization process itself is unsupervised, but viewed as part of the full training pipeline, the overall method is semi-supervised.

This paper provides a theoretical answer. We show that these methods all optimize the same objective: *coherence*, defined as the joint likelihood of the model's behaviors across all contexts, with the joint likelihood in turn defined through autoregressive conditioning or training. We prove that coherence optimization is equivalent to description-length regularization, and that it is the optimal regularization scheme for semi-supervised learning when the regularizer is derived from a pretrained model.

The Problem (Semi-Supervised Learning). Consider a model responding to different contexts (e.g., question prompts), where each context is associated with a large space of possible behaviors (e.g., free-form answers). A *deterministic policy* (d-policy) is a complete assignment of one behavior to each context. Given N supervised labels, we want to find the d-policy that generalizes best to unseen contexts. The classical solution is empirical risk minimization (ERM), which fits the supervised labels exactly, but ERM overfits when the hypothesis space is large, as is the case for language models (Lampinen et al., 2025). By utilizing access to unlabeled contexts, we would like to improve the generalizability of the learned hypothesis.

The Solution (Coherence as Regularization). Statistical learning theory addresses overfitting through regularization that penalizes complex hypotheses (Vapnik, 1999). Description-length regularizers, which penalize hypotheses that are hard to compress under a prior distribution, form a classical and general family within this framework. The most famous such prior is the Solomonoff prior (Solomonoff, 2009). We prove

[1] Specifically, Theorem 5.5 shows that self-improvement succeeds when the prior is a good approximation to the data-generating distribution, and Conjecture 5.9 identifies when purely unsupervised coherence optimization recovers the regularized training objective.

[2] Connections and differences between ICM and coherence optimization is discussed in §2.

that among all description-length regularizers for semi-supervised learning, the optimal one uses the prior that best approximates the data-generating distribution (Theorem 5.5). For language models, this is the pretrained model. The description length under this prior equals the negated coherence. Empirical results support this conclusion.

Formally speaking, we prove that among all description-length regularization schemes for semi-supervised learning, coherence regularization derived from a KL-optimal prior is optimal in the following sense. It maximizes the worst-case lower bound on expected accuracy (Theorem 5.5). Since the optimal prior is the one that minimizes KL divergence to the data-generating distribution, and a pretrained language model is the best practically obtainable such approximation (having been trained on the largest available sample), pretrained models provide the tightest worst-case accuracy guarantee among available priors.

The Algorithm (Gibbs Sampling). Direct optimization of coherence is intractable because it requires evaluating the joint probability over exponentially many behavior combinations. We show that Gibbs sampling provides an efficient solution. The algorithm repeatedly (i) selects a context, (ii) resamples its behavior conditioned on all other current behaviors, and (iii) updates the d-policy. Under mild conditions, this converges to a distribution concentrated on high-coherence d-policies (Theorem 4.2).

Contributions. We make three main contributions:

1. We show that debate, bootstrap, and ICM are all special cases of coherence optimization.
2. We prove that coherence optimization is the optimal form of description-length regularization for semi-supervised learning with a pretrained prior. Preliminary experiments show that coherence-based regularizers outperform LLM-as-a-judge as proxies for truthfulness.
3. We present a simple, scalable, and theoretically grounded algorithm for general coherence optimization.

2 Related Works

This section reviews literature on the formal and experimental study of feedback-free self-improvement.

Unsupervised Scalable Oversight and Feedback-Free Self-Improvement. Scalable oversight (Bowman et al., 2022), an area of AI alignment research (Ji et al., 2025), focuses on directly or indirectly supervising strong and potentially superhuman models, ones capable of gaming any available supervision signal. Classical methods of scalable oversight include debate (Irving et al., 2018; Brown-Cohen et al., 2024; Khan et al., 2024), iterated amplification (Wu et al., 2021), recursive reward modeling (Leike et al., 2018), and weak-to-strong generalization (Kenton et al., 2024). Some of these methods, including debate, are unsupervised or self-supervised, where the aim is to uplift the capabilities of a model using its own supervision, without sacrificing safety properties. Newer methods in this category, including internal coherence maximization (Wen et al., 2025) and sampling methods based on Markov chain Monte Carlo (Karan & Du, 2025), have managed to match supervised finetuning performance on golden labels, a surprising feat for unsupervised methods. To date, the study of such methods has remained heuristic and entirely empirical — with rare exceptions studying the computational complexity of debate (Brown-Cohen et al., 2024; 2025) — and it is generally not understood why such methods work. This paper aims to change that by formally characterizing the mechanism of such feedback-free self-improvement (i.e., that these methods are policy-wide description-length regularization), and proves that coherence is the optimal regularization scheme for semi-supervised learning. It also gives a practical and easily scalable algorithm and demonstrates its empirical promise. The ICM algorithm (Wen et al., 2025) is closely related to the formalism of coherence optimization. It optimizes for the bidirectional variant of the autoregressively defined objective in coherence optimization, as will be discussed in §4.2. It is exactly this difference, however, that enables the scalable optimization and formal analysis of the autoregressive objective in this paper, which was difficult or intractable for ICM.

Theory of Semi-Supervised Learning and Regularization. The theory of semi-supervised learning characterizes how knowledge of the marginal distribution $P(x)$ of the input x restricts the effective complexity

of learning the conditional output distribution $P(y|x)$. Consistency regularization is a classical type of semi-supervised learning method, shown to reduce the local Rademacher complexity of the hypothesis class when applied on unlabeled data and lead to tighter generalization bounds (Maximov et al., 2018). While it is famously shown that unlabeled data improves sample complexity by at most a constant factor in the worst case (Ben-David et al., 2008), that constant factor depends on the data dimension and can be large in high-dimensional regimes. For instance, in sparse Gaussian mixtures, unlabeled data enables polynomial-time learning where supervised methods are computationally intractable (Azar & Nadler, 2024), and in the case of large neural networks, the neural scaling law varies with the intrinsic dimension of the data manifold (Sharma & Kaplan, 2022). However, high dimensionality can also be a curse; dimensional collapse — where embeddings degenerate into a low-rank subspace due to implicit regularization — is a common failure mode for semi-supervised learning (Jing et al., 2022), which He et al. (2024) propose countering via orthogonality regularization. In the deep learning regime, consistency regularization is shown to be equivalent to spectral decomposition of the augmentation graph (HaoChen et al., 2021). Wei et al. (2021) provided guarantees for self-training under an expansion assumption, a condition recently adapted to explain how strong student models can generalize beyond weak supervisors (Lang et al., 2024) . At the information-theoretic limit, these regularizers can be modeled with Solomonoff induction (Leike, 2016), a connection formalized with singular learning theory to link loss landscape degeneracy to model compressibility (Urdshals et al., 2025). Our work contributes to this literature by (i) generalizing consistency regularization from invariance under local ϵ-perturbations to predictability under arbitrary context differences; and (ii) suggesting that in the high-dimensional setting of language models, coherence regularization optimally reaps the large, dimensionality-dependent improvement factor from unlabeled data access, as upper-bounded by Ben-David et al. (2008).

Formal Accounts of Reflective Equilibrium. Interestingly, another line of related work comes from philosophical epistemology, which gives a conceptual and formal description of the possible *end states* of coherence optimization. Reflective equilibrium, first introduced in Rawls (1971), refers to the process of iteratively revising one's principles and case judgments to remove inconsistencies between them, until eventually converging upon the state of equilibrium. While the original proposal cleanly distinguishes principles from judgments, there have been generalized proposals considering beliefs lying on the full spectrum of generality (Rawls, 2005). Attempts to formalize reflective equilibrium with the language of AI started with Beisbart (2021), which built a formal model of reflective equilibrium with a mix of classical logic and optimization formalism, while stochastic elements were later introduced in Dellsén (2024). Prompted by theoretical interest in the question of when reflective equilibria of individuals converge in a group (Tersman, 2024), simulation studies have emerged with a diversity of setup design, including in the space of formal logic with logical inference distances as proximity measure (Lohse, 2023) and in the space of simple classifiers with classification margins as proximity measure (Baumgaertner & Lassiter, 2024). There also exists theoretical debate on, to paraphrase in machine learning language, whether the mutual support in a reflective equilibrium should be autoregressive (Daniels, 1996; Holmgren, 1989) or bidirectional (DePaul, 2006; Haslett, 1987), with concerns that the latter leads to circularity; we revisit this question in §4, when comparing ICM and coherence optimization. All the works above have been limited to highly simplistic belief spaces and usually without any element of *learning*, which makes such formalisms inappropriate for actual application in machine learning. We aim to change this by introducing an expressive, learning-based formalism that comes directly with practical algorithms implemented on large language models, and provides grounding for reflective equilibrium by associating it, both theoretically and experimentally, with increased accuracy and truthfulness.

3 Defining Coherence

We develop a unified framework for understanding coherence optimization as optimal regularization in semi-supervised learning. We begin with the abstract problem, then introduce learning systems as a tractable way to implement the optimal solution.

3.1 The Semi-Supervised Learning Problem

Consider a semi-supervised learning setup with behavior space $\mathcal{A}$ (a finite set of all possible behaviors) and context partition $\mathcal{S} \in \Pi(\mathcal{A})$ (a partition of $\mathcal{A}$ into sets of competing behaviors). Each context $s \in \mathcal{S}$ is a set of mutually exclusive behaviors, and each behavior $a \in \mathcal{A}$ belongs to exactly one context.

Definition 3.1 (Deterministic Policy)**.** A *deterministic policy* (d-policy) is a function $\pi : \mathcal{S} \to \mathcal{A}$ such that $\pi(s) \in s$ for all $s \in \mathcal{S}$. The space of all d-policies is $\mathcal{A}^{\mathcal{S}} \coloneqq \prod_{s \in \mathcal{S}} s$.

Let $\mathcal{D} \in \Delta\left[\mathcal{A}^{\mathcal{S}}\right]$ be the data-generating distribution over d-policies. Let $\mathcal{F} \in \Delta[\mathcal{S}]$ be the context distribution, and $\{(s_n, a_n)\}_{n=1}^{N}$ the supervised samples (context-behavior pairs with ground-truth labels). We assume unlimited access to unsupervised samples (contexts in $\mathcal{F}$, without labels).

Our goal is to find a d-policy $\pi \in \mathcal{A}^{\mathcal{S}}$ that maximizes accuracy:

$$\alpha(\pi) \coloneqq \Pr_{s \sim \mathcal{F}, \pi^* \sim \mathcal{D}}\left[\pi(s) = \pi^*(s)\right].$$

Empirical risk minimization (ERM) overfits when the hypothesis space is too large and expressive. The classical remedy is structural risk minimization (SRM) with description-length regularization. Given a prior distribution $\mathcal{P} \in \Delta\left[\mathcal{A}^{\mathcal{S}}\right]$ over d-policies,

$$\begin{aligned} \pi_{\text{SRM}} &\coloneqq \operatorname*{arg\,max}_{\pi \in \mathcal{A}^{\mathcal{S}}} \ \alpha_{\text{train}}(\pi) - \text{reg}(\pi), \text{ where} \\ \alpha_{\text{train}}(\pi) &\coloneqq \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{\pi(s_n) = a_n}, \\ \text{reg}(\pi) &\coloneqq \left(\frac{-2 \log_2 \mathcal{P}(\pi) + \log_2 e - \log_2(1/\delta)}{2N} \right)^{1/2}, \end{aligned} \tag{1}$$

and $\mathcal{P}(\pi)$ is the probability mass of d-policy π under prior $\mathcal{P}$.

We will show in §5 that the optimal choice of prior $\mathcal{P}$ for SRM is the one that minimizes $\text{KL}[\mathcal{D} \| \mathcal{P}]$, i.e., the best available approximation to the true data-generating distribution. For language models, this is the pretrained model. §5.2 formalizes this intuition and proves that coherence regularization with a pretrained prior is optimal among all description-length regularizers in a worst-case asymptotic sense.

3.2 Learning Systems as Tractable Priors

The optimal prior $\mathcal{P}$ is the best approximation to the data-generating distribution, but we still need to compute probabilities under it. A prior over d-policies assigns probability to each of the $\prod_{s \in \mathcal{S}} |s|$ possible d-policies, and direct enumeration is intractable.

We need priors that allow efficient computation. A *learning system* $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$ defines such a prior, where $\mathcal{M}$ is a space of stochastic policies (e.g., training datasets or in-context example sets) and σ is an inference function. The learning system specifies how to compute d-policy probabilities autoregressively, as the probability of behavior a_1 in the context s_1, times the probability of a_2 in s_2 conditioned on (s_1, a_1), and so on. The log of this joint probability thereby equals what we call the *coherence* of the d-policy. By optimizing for coherence, we maximize $\log_2 \mathcal{P}(\pi)$, and thereby minimize $\text{reg}(\pi)$ in Equation 1. For now, we will thus focus on characterizing and optimizing for coherence.

Definition 3.2 (Learning System)**.** A learning system is a quadruple $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$, where the *policy monoid* $\langle \mathcal{M}, + \rangle$ is a commutative monoid with the identity element 0 (the *base policy*), a finite generating set $\mathcal{A} \subset \mathcal{M}$ of $\mathcal{M}$ (the *behavior space*), a partition $\mathcal{S} \in \Pi(\mathcal{A})$ of the behavior space $\mathcal{A}$ (the *context partition* or the *competing behavior partition*), and the *inference function* σ mapping every $(\phi, s) \in \mathcal{M} \times \mathcal{S}$ to a distribution

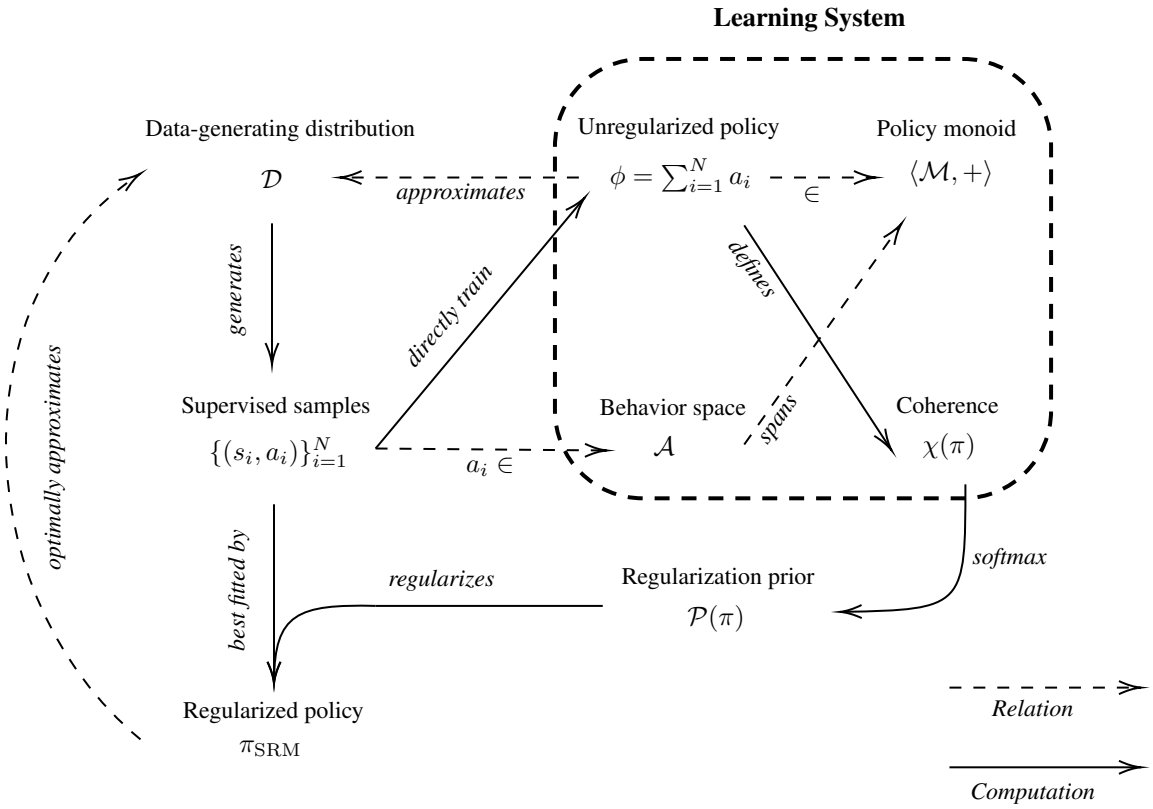

Figure 1: The coherence optimization framework. The data-generating distribution $\mathcal{D}$ produces supervised samples. Coherence optimization finds π_{SRM} maximizing empirical accuracy with regularization from prior $\mathcal{P}$. A learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$ defines the coherence function χ, providing a tractable instance of $\mathcal{P}$ via softmax over coherence: $\mathcal{P}(\pi) = \mathrm{X}^{\beta}(\pi) \propto 2^{\beta\chi(\pi)}$. Bayesian inference, in-context learning (ICL), and finetuning are instances of learning systems.

over s that satisfies the *chain rule*:

$$\sigma(\phi, s_1)(a_1) \cdot \sigma(\phi + a_1, s_2)(a_2) = \sigma(\phi, s_2)(a_2) \cdot \sigma(\phi + a_2, s_1)(a_1), \quad \forall a_1 \in s_1 \in \mathcal{S}, a_2 \in s_2 \in \mathcal{S}.$$

The notation in Definition 3.2 directly extends §3.1. The behavior space $\mathcal{A}$ and context partition $\mathcal{S}$ are the same objects, and a d-policy $\pi \in \mathcal{A}^{\mathcal{S}}$ satisfies $\pi(s) \in s$ for all $s \in \mathcal{S}$. The main purpose of learning systems is to provide a tractable class of priors $\mathcal{P}$ for SRM, namely via the *softmax over coherence* (defined below).

Intuitively, every element in the behavior space is a "sample" with a certain input-output behavior, and every stochastic policy $\phi \in \mathcal{M}$ can be represented by a multiset of samples, i.e., its "training data". Each context $s \in \mathcal{S}$ is a set of competing behaviors. The chain rule asks that the cumulative "loss" of sequentially training on multiple samples is not affected by the ordering of samples.

Example 3.3 (Bayesian Learning System)**.** A Bayesian learning system is defined by

$\langle \mathcal{M}, + \rangle$ Let a latent variable $\theta \in \Theta$ govern the data-generating process. A stochastic policy $\phi \in \mathcal{M}$ corresponds to a posterior distribution $\Pr[\theta \mid D_\phi]$ over Θ, obtained by conditioning a base prior $\Pr[\theta]$ on a multiset of observations D_ϕ. The base policy 0 corresponds to the prior $\Pr[\theta]$ (i.e., $D_0 = \emptyset$). The commutative monoid $\langle \mathcal{M}, + \rangle$ is the space of these observation multisets, with multiset union as the operation $+$.

$\mathcal{A}$ The set of all possible observations or data points.

$\mathcal{S}$ A partition of $\mathcal{A}$ where each context $s \in \mathcal{S}$ represents a set of mutually exclusive outcomes for a single experiment or query.

σ The Bayesian predictive distribution. For a stochastic policy ϕ (representing data D_ϕ) and a context s, the probability of observing a specific behavior $a \in s$ is its marginal likelihood: $\sigma(\phi, s)(a) = \Pr[a \mid D_\phi] = \int_{\theta \in \Theta} \Pr[a \mid \theta] \Pr[\theta \mid D_\phi] \mathrm{d}\theta$.

The chain rule holds as a direct consequence of the definition of joint probability. The LHS is

$$\sigma(\phi, s_1)(a_1) \cdot \sigma(\phi + a_1, s_2)(a_2) = \Pr[a_1 \mid D_\phi] \Pr[a_2 \mid D_\phi, a_1] = \Pr[a_1, a_2 \mid D_\phi],$$

while the RHS is

$$\sigma(\phi, s_2)(a_2) \cdot \sigma(\phi + a_2, s_1)(a_1) = \Pr[a_2 \mid D_\phi] \Pr[a_1 \mid D_\phi, a_2] = \Pr[a_2, a_1 \mid D_\phi].$$

Since joint probability is symmetric, the rule holds.

Example 3.4 (In-Context Learning System)**.** An in-context learning system is defined by

$\langle \mathcal{M}, + \rangle$ The policy space $\mathcal{M}$ is the set of all possible multisets of in-context examples. The base policy 0 is the empty multiset. The operation $+$ is multiset union, which is commutative by definition. In practice, these multisets are linearized into sequences to be fed into a language model, and this linearization breaks strict commutativity; hence, this is an approximation.

$\mathcal{A}$ The set of individual input-output pairs that can be used as in-context examples, e.g., (context, behavior) pairs.

$\mathcal{S}$ Each context $s \in \mathcal{S}$ is the set of possible behaviors for a given input. For instance, for an input q_s, $s = \{(q_s, a) \mid a \in \text{possible behaviors}\}$.

σ For a stochastic policy $\phi \in \mathcal{M}$ (a multiset of examples) and a context s corresponding to input q_s, $\sigma(\phi, s)(a)$ for $a = (q_s, \tilde{a}) \in s$ is the probability assigned by the language model to generating behavior $\tilde{a}$ given the prompt constructed from ϕ and q_s. That is, $\sigma(\phi, s)(a) = P_{\mathrm{LM}}(\tilde{a} | \mathrm{context}(\phi), q_s)$.

Note that this example defines in-context learning as a *learning system*, which provides a tractable prior for regularization. It does not claim that in-context learning is itself coherence optimization. The claim in §4.2 is that methods like debate, bootstrap, and ICM optimize coherence as defined *through* such learning systems.
For in-context learning systems, it is unclear how well the commutativity of $\langle \mathcal{M}, + \rangle$ and the chain rule hold in practice. However, to the extent that in-context learning is an approximation of Bayesian inference (Xie et al., 2021), such assumptions make sense as a first-order approximation. Also, in practice, we only face in-context sequences that are relatively well-mixed, so order dependency issues caused by a big in-context "distribution shift" over the turns is not a severe concern here.

Example 3.5 (Finetuning Learning System)**.** A finetuning learning system is defined by...

$\langle \mathcal{M}, + \rangle$ The policy space $\mathcal{M}$ can be identified with the set of all possible multisets of training data. The base policy 0 corresponds to the empty dataset and a randomly initialized model. The operation $+$ is multiset union. A stochastic policy $\phi \in \mathcal{M}$ represents the dataset used to train a model from its initial state to its current state.

$\mathcal{A}$ The set of all possible labeled data points (x, y).

$\mathcal{S}$ Each context $s \in \mathcal{S}$ corresponds to a single input x_s and is the set of all possible labeled data points for that input: $s = \{(x_s, y) \mid y \in \text{possible labels}\}$.

σ For a stochastic policy $\phi \in \mathcal{M}$ (representing training data D_ϕ) that has produced a model with parameters θ_ϕ, and a context s corresponding to input x_s, $\sigma(\phi, s)(a)$ for $a = (x_s, y) \in s$ is the probability the model assigns to label y for input x_s. That is, $\sigma(\phi, s)(a) = P(y|x_s; \theta_\phi)$.

As with Example 3.4, this example defines finetuning as a learning system (a source of tractable priors), not as a form of coherence optimization itself.
For finetuning learning systems, it is unclear how well the commutativity of $\langle \mathcal{M}, + \rangle$ and the chain rule hold in practice. For instance, the order of training data matters for stochastic gradient descent. However, to the extent that supervised learning is an approximation of Bayesian inference (Mingard et al., 2021), such assumptions make sense as a first-order approximation. Also, as before, we only face training example sequences that are relatively well-mixed, so order dependency issues caused by big distribution shifts are not a severe concern here.

Remark 3.6 (On the Chain Rule Assumption). A formal bound on the approximation error of the chain rule for in-context learning and finetuning systems remains an open question. However, the practical algorithms for coherence optimization do not depend on the chain rule holding exactly. The Gibbs sampling algorithm (Algorithm 1) optimizes conditional predictability across contexts, and the mutual predictability objective (Equation 3) uses bidirectional conditioning. Both remain well-defined and meaningful even when the chain rule is only approximately satisfied. In particular, they do not reduce to greedy decoding, because they optimize joint predictability across the full set of contexts rather than marginal likelihood of individual behaviors. The chain rule is needed for the equivalence between coherence and joint log-probability (Definition 3.2), but the algorithms optimize conditional predictability directly and converge to high-quality d-policies regardless.

D-policies $\pi \in \mathcal{A}^{\mathcal{S}}$ are to be distinguished from stochastic policies $\phi \in \mathcal{M}$. The latter is probabilistic, mapping every context to a distribution over its behaviors via the inference function σ.

We now define coherence, the central quantity of this paper. Coherence measures how well the responses in a d-policy "fit together" according to the prior policy. A high-coherence d-policy is one where each response is predictable given the others.

Definition 3.7 (Coherence)**.** Given any *prior policy* $\phi \in \mathcal{M}$, for any d-policy π, its *coherence* $\chi_\phi(\pi)$ relative to ϕ is defined by

$$\chi_\phi(\pi) = \sum_{n=1}^{|\mathcal{S}|} \log_2 \sigma\left(\phi + \sum_{m=1}^{n-1} \pi(s_m), s_n\right)\big(\pi(s_n)\big) \leq 0$$

where $s_1, \cdots, s_{|\mathcal{S}|}$ are the contexts in $\mathcal{S}$ in arbitrary order. The chain rule ensures that $\chi_\phi(\pi)$ is well-defined, as it implies that one can always swap neighbouring contexts in the sequence $\{s_i\}$ without changing $\chi_\phi(\pi)$, and thus the ordering of $\{s_i\}$ does not affect $\chi_\phi(\pi)$. In particular, we denote $\chi(\pi) \coloneqq \chi_0(\pi)$.

Note that coherence operates on the probabilities assigned by the inference function σ, not on string equality. If the prior model has learned that two behaviors are semantically equivalent (e.g., $P[\text{yes} \mid \text{yeah}] \approx P[\text{yes} \mid \text{yes}]$), they will receive similar conditional probabilities and thus contribute similarly to coherence. Semantic equivalence is handled to the extent that the prior policy treats semantically equivalent outputs similarly.

From now on, we will denote with $s_1, \cdots, s_{|\mathcal{S}|}$ an arbitrary ordering of contexts in $\mathcal{S}$.

Definition 3.8 (Softmax Over Coherence)**.** For any *temperature reciprocal* $\beta > 0$, the *softmax over coherence* $\mathrm{X}^\beta : \mathcal{A}^\mathcal{S} \to \mathbb{R}_{\geq 0}$ is a probability distribution over d-policies such that the probability mass $\mathrm{X}^\beta(\pi) \propto 2^{\beta\chi(\pi)}$. In particular, when $\beta = +\infty$, X^β collapses into a uniform distribution over $\arg\max_\pi \chi(\pi)$, with 0 probability mass everywhere else.

The softmax over coherence at $\beta = 1$ provides an instance of the prior $\mathcal{P}$ from §3.1. Setting $\mathcal{P} = \mathrm{X}^1$ gives $\mathcal{P}(\pi) \propto 2^{\chi(\pi)}$. This is the tractable prior that learning systems provide for description-length regularization.

Example 3.9 (Sauces)**.** Consider the following learning system. The context partition is $\mathcal{S} = \{s^{\text{burger}}, s^{\text{fries}}\}$, where $s^{\text{burger}} = \{a^{\text{burger}}_{\text{mayo}}, a^{\text{burger}}_{\text{mustard}}, a^{\text{burger}}_{\text{other}}\}$ contains behaviors in the context "What sauce do you use for burgers?" and $s^{\text{fries}} = \{a^{\text{fries}}_{\text{mayo}}, a^{\text{fries}}_{\text{ketchup}}, a^{\text{fries}}_{\text{other}}\}$ contains competing behaviors in the context "What sauce do you use for dipping French fries?"

Consider the Bayesian learning system $\mathcal{M}$ with the following prior over $s^{\text{burger}} \times s^{\text{fries}}$:

	$a^{\text{burger}}_{\text{mayo}}$	$a^{\text{burger}}_{\text{mustard}}$	$a^{\text{burger}}_{\text{other}}$
$a^{\text{fries}}_{\text{mayo}}$	0.3	ϵ	ϵ
$a^{\text{fries}}_{\text{ketchup}}$	ϵ	$0.175 - \epsilon$	$0.175 - \epsilon$
$a^{\text{fries}}_{\text{other}}$	ϵ	$0.175 - \epsilon$	$0.175 - \epsilon$

Given the symmetry, let us only calculate X^β for the two d-policies $\pi_1 = (a^{\text{burger}}_{\text{mayo}}, a^{\text{fries}}_{\text{mayo}})$ and $\pi_2 = (a^{\text{burger}}_{\text{mustard}}, a^{\text{fries}}_{\text{ketchup}})$. For simplicity, assume $\epsilon = 0$.

In a Bayesian learning system, the coherence $\chi(\pi)$ simplifies to the $\log_2$ joint probability of the d-policy under the prior. Let $s_1 = s^{\text{burger}}$ and $s_2 = s^{\text{fries}}$. Then

$$\begin{aligned}
\chi(\pi) &= \log_2 \sigma(0, s_1)(\pi(s_1)) + \log_2 \sigma(\pi(s_1), s_2)(\pi(s_2)) \\
&= \log_2 P(\pi(s_1)) + \log_2 P(\pi(s_2) | \pi(s_1)) \\
&= \log_2 \left(P(\pi(s_1)) \cdot \frac{P(\pi(s_1), \pi(s_2))}{P(\pi(s_1))} \right) \\
&= \log_2 P(\pi(s_1), \pi(s_2)).
\end{aligned}$$

Using the provided table with $\epsilon = 0$: for $\pi_1 = (a^{\text{burger}}_{\text{mayo}}, a^{\text{fries}}_{\text{mayo}})$, the joint probability is $P(a^{\text{burger}}_{\text{mayo}}, a^{\text{fries}}_{\text{mayo}}) = 0.3$. Thus, its coherence is $\chi(\pi_1) = \log_2(0.3) \approx -1.74$. For $\pi_2 = (a^{\text{burger}}_{\text{mustard}}, a^{\text{fries}}_{\text{ketchup}})$, the joint probability is $P(a^{\text{burger}}_{\text{mustard}}, a^{\text{fries}}_{\text{ketchup}}) = 0.175$. Thus, its coherence is $\chi(\pi_2) = \log_2(0.175) \approx -2.51$.

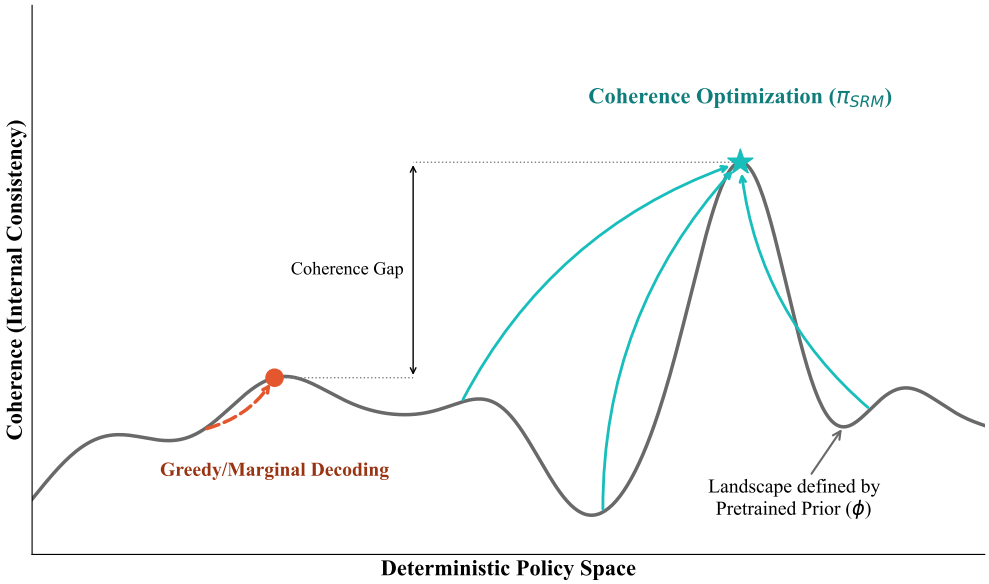

Figure 2: While greedy decoding gets trapped in local optima, coherence optimization finds the global peak of coherence/compressibility defined by the pretrained prior. Note the distinction between the coherence *landscape* (determined by the pretrained prior) from the *points* (deterministic policies in $\mathcal{A}^{\mathcal{S}}$).

We see that $\chi(\pi_1) > \chi(\pi_2)$. This means the d-policy of using mayo for both burgers and fries is more coherent, even though mayo is not the most popular marginal choice for either food item. The marginal probabilities are $P(a^{\text{burger}}_{\text{mayo}}) = 0.3$ vs. $P(a^{\text{burger}}_{\text{mustard}}) = 0.35$, and $P(a^{\text{fries}}_{\text{mayo}}) = 0.3$ vs. $P(a^{\text{fries}}_{\text{ketchup}}) = 0.35$.

Remark 3.10 (What is coherence?). The sauces example might make coherence seem trivial at first. Why do we need another fancy name for joint probability? The value of such concept will be much clearer in an in-context learning system, and, even more so, in a finetuning learning system. In these setups, coherence is no longer reducible to well-known existing concepts and is hard to optimize for directly.

For an in-context learning system, coherence equals the cumulative perplexity when generating a list of d-policy responses *sequentially*. Note that greedy decoding at temperature 0 does *not* optimize such a cumulative perplexity (neither in theory nor in practice); the only decoding method that does it in theory is beam search with an infinite number of beams.

For a finetuning learning system, coherence equals the cumulative cross-entropy loss when training the prior policy on a list of d-policy responses *sequentially*. Despite the definition, however, it's almost certainly intractable to directly perform gradient descent on such a coherence function to optimize for it.

Given these challenges in optimizing for coherence, how should we proceed?

4 Algorithms for Coherence Optimization

Having defined coherence, we now turn to the question of its optimization. Figure 2 illustrates the goal of coherence optimization.

4.1 Gibbs Sampling for Coherence Optimization

We introduce an efficient algorithm for finding the d-polic(ies) that maximize coherence.[3]

[3]In Algorithm 1, the 'β' in σ^β means applying the transformation $p \to p^\beta$ to all probability masses and re-normalizing them.

Algorithm 1 Gibbs Sampling Over D-Policies

1: **Input:** Learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$; initial d-policy π_0; number of steps N; temperature reciprocal β.
2: **Output:** List of d-policies $\pi_0, \pi_1, \cdots, \pi_N$.
3: **procedure** GibbsSample
4: **for** $t = 0, 1, \ldots, N-1$ **do**
5: Sample $n \sim \mathrm{Unif}[1..|\mathcal{S}|]$.
6: Sample $a_t^n \sim \sigma^\beta\left(\sum_{m \neq n} \pi_t(s_m), s_n\right)$.
7: Set π_{t+1} such that $\pi_{t+1}(s_n) = a_t^n$ and $\pi_{t+1}(s_m) = \pi_t(s_m)$ for $m \neq n$.
8: **end for**
9: **end procedure**

The algorithm is simple to implement (repeatedly resampling behaviors using behaviors in other contexts as condition) and easy to scale (see Algorithm 2 for a variant optimized for concurrency). A variant of it was implemented by Xu et al. (2023) for the different purpose of learning Chain-of-Thought reasoning.

We show below that Algorithm 2 approximately maximizes coherence.

Definition 4.1 (Ergodic Learning System)**.** A learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$ is *ergodic* if for every proper subset C of the d-policy space $\mathcal{A}^\mathcal{S}$, there exist d-policies $\pi \in C$ and $\pi' \in \mathcal{A}^\mathcal{S} \setminus C$ such that π' agrees with π under all contexts except one $\dot{s} \in \mathcal{S}$, and $\sigma(\sum_{s \neq \dot{s}} \pi(s), \dot{s})(\pi'(\dot{s})) > 0$. In particular, all Bayesian learning systems with a full-support prior are ergodic (see Appendix B for proof).

Theorem 4.2 (Gibbs Sampling Recovers Softmax Over Coherence)**.** *For ergodic learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$, initial d-policy π_0, and temperature reciprocal $\beta > 0$, when $N \to +\infty$, we have, for Algorithm 1,*

$$\pi_r \xrightarrow{d} \mathrm{X}^\beta, \quad \textit{where } r \sim \mathrm{Unif}(\{0, \cdots, N\}).$$

Proof. See Appendix B. □

4.2 Reductions from Existing Methods of Self-Improvement

Having formulated a general and tractable algorithm for coherence optimization, we now show that state-of-the-art feedback-free self-improvement methods are reducible to coherence optimization.

Debate. Debate (Irving et al., 2018) is an unsupervised method for scalable oversight where copies of a language model argue for opposing propositions in a back-and-forth structure. It has been shown to significantly increase language model accuracy on factual tasks (Li et al., 2022; Du et al., 2023), and allows for effective weak-to-strong truthfulness supervision (Khan et al., 2024).

The connection to coherence optimization is direct. Debate is Gibbs sampling with $|\mathcal{S}| = 2$. Consider a learning system with two contexts, $\mathcal{S} = \{s_{\text{pro}}, s_{\text{con}}\}$, representing arguments for and against a proposition P. The iterative debate process, where each side updates its argument based on the other's, is formally identical to Algorithm 1. Each debater resamples their argument conditioned on their opponent's current argument, exactly as Gibbs sampling resamples one component of the d-policy conditioned on all others.

Proposition 4.3 (Debate Maximizes Coherence)**.** *Iterative debate for a learning system with $|\mathcal{S}| = 2$ is a synchronous variant of Gibbs Sampling (Algorithm 1). For an ergodic learning system, the distribution of d-policies π_t converges to the softmax over coherence X^β.*

Appendix C.1 gives a formal specification and algorithm pseudocode of iterative debate. Proposition 4.3 then follows by tautology.

Simple Bootstrap. Bootstrap methods, where the output of a model is used for training itself, have been reported to increase accuracy without supervision as early as in Lee et al. (2013). It is recently revisited in the language modeling context as an unsupervised *inference-time* technique, where it matches the performance of supervised finetuning on golden labels (Chen et al., 2023).

Simple bootstrap can be seen as an approximation of Gibbs sampling where history accumulates rather than being replaced. Instead of maintaining a fixed-size d-policy and resampling components, simple bootstrap sequentially generates behaviors $a_1, a_2, a_3, \ldots$, where the context for generating a_n is the entire preceding sequence.

Proposition 4.4 (Simple Bootstrap Approximates Coherence)**.** *Let the sequence of contexts $(s_1, \ldots, s_N)$ be a permutation of $\mathcal{S}$ chosen uniformly at random. The total variation distance $D_{\mathrm{TV}}(P_{\mathrm{SB}}, \mathrm{X}^{\beta})$ between the simple bootstrap distribution and the softmax over coherence satisfies $D_{\mathrm{TV}} = O(|\beta - 1|)$ as $\beta \to 1$.*

Proof. See Appendix C.2. □

The $O(|\beta - 1|)$ approximation error limits the usefulness of this approximation at low temperatures ($\beta \gg 1$), where Gibbs sampling remains effective but simple bootstrap may not.

Internal Coherence Maximization. Internal coherence maximization (ICM) (Wen et al., 2025) optimizes for mutual predictability between model responses on different contexts through a hill-climbing process. Like simple bootstrap, it matches the performance of golden label finetuning.

Coherence and mutual predictability (the ICM objective) are closely related:

$$\chi(\pi) = \sum_{n=1}^{|\mathcal{S}|} \log_2 \sigma\left(\sum_{m<n} \pi(s_m), s_n\right)(\pi(s_n)), \tag{2}$$

$$f_{\mathrm{MP}}(\pi) = \sum_{n=1}^{|\mathcal{S}|} \log_2 \sigma\left(\sum_{m\neq n} \pi(s_m), s_n\right)(\pi(s_n)). \tag{3}$$

The difference is that coherence uses autoregressive conditioning ($m < n$) while mutual predictability uses bidirectional conditioning ($m \neq n$). Bidirectional conditioning is more expensive to compute, which is why ICM is restricted to domains with few discrete responses (e.g., multiple-choice questions), while Gibbs sampling over coherence scales to open-ended generation. See Appendix C.3 for further analysis.

5 Optimality of Coherence Optimization

In this section, we establish that coherence optimization is a form of description-length regularization, and that it is, in a worst-case asymptotic sense, the optimal form of description-length regularization.

5.1 Coherence as Description-Length Regularization

The concept of coherence has a natural and powerful interpretation, specifically the principle of minimum description length (MDL). The MDL principle states that the best model for a set of data is the one that permits the shortest description of the data.

Consider a d-policy π as a dataset, where each $\pi(s_n)$ is a data point. The coherence $\chi(\pi)$ is defined as a sum of sequential log-probabilities:

$$\chi(\pi) = \log_2 \sigma(0, s_1)(\pi(s_1)) + \log_2 \sigma(\pi(s_1), s_2)(\pi(s_2)) + \ldots$$

This is precisely the log-probability of observing the sequence of behaviors $(\pi(s_1), \pi(s_2), \ldots)$ under the predictive model defined by the learning system. By the chain rule, this is equal to the joint log-probability,

$\log_2 \Pr[\pi(s_1), \ldots, \pi(s_{|\mathcal{S}|})]$. By arithmetic coding, negated coherence is, when rounded up, the *number of bits needed to describe the d-policy.*

Coherence optimization, therefore, acts as a form of regularization that favors d-policies embodying strong, compressible patterns, and forces the model to discover and adhere to underlying principles rather than memorizing a set of unrelated behaviors. We can thus prove that coherence optimization, like other types of regularization, gives a uniform convergence guarantee on the d-policy's generalization error.

Definition 5.1 (Agreement and Accuracy)**.** For any two d-policies $\pi_1, \pi_2 \in \mathcal{A}^{\mathcal{S}}$ and a subset of contexts $\hat{\mathcal{S}} \subseteq \mathcal{S}$, we define the *agreement* between π_1 and π_2 on $\hat{\mathcal{S}}$ as

$$\alpha_{\hat{\mathcal{S}}}(\pi_1; \pi_2) = \alpha_{\hat{\mathcal{S}}}(\pi_2; \pi_1) \coloneqq \frac{1}{|\hat{\mathcal{S}}|} \sum_{s \in \hat{\mathcal{S}}} \mathbf{1}_{\pi_1(s) = \pi_2(s)}.$$

In particular, we denote $\alpha(\pi_1; \pi_2) \coloneqq \alpha_{\mathcal{S}}(\pi_1; \pi_2)$. Given any *ground-truth d-policy* π^*, we define d-policy π's *accuracy* on $\hat{\mathcal{S}}$ as $\alpha_{\hat{\mathcal{S}}}(\pi; \pi^*)$.

Theorem 5.2 (Uniform Convergence)**.** *Let π^* be any given ground-truth d-policy, and the* training contexts *$\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_N$ be uniformly and independently sampled contexts from $\mathcal{S}$, for some $N > 0$. Let π be an arbitrary d-policy with $\alpha_{\text{train}}(\pi; \pi^*) \coloneqq \alpha_{\{\hat{s}_i\}_{i=1}^N}(\pi; \pi^*)$, then*

$$|\alpha(\pi; \pi^*) - \alpha_{\text{train}}(\pi; \pi^*)| \leq \sqrt{\frac{-2\chi(\pi) + \log_2 e - \log_2(1/\delta)}{2N}}$$

holds uniformly for all π with probability $1 - \delta$, for any given $\delta \in (0, 1)$.

Theorem 5.2 tells us that, the coherence of a d-policy helps bound its generalization gap, i.e., the difference between its training and actual accuracy. But recall that coherence can be defined with respect to any prior policy; does the choice of prior policy affect the generalization gap? The answer is yes.

Definition 5.3 (Optimality Gap)**.** Given any ground-truth d-policy π^*, for any stochastic policy $\phi \in \mathcal{M}$, we define the *optimality gap* of ϕ as the prior policy to be

$$G(\phi; \pi^*) \coloneqq -2\chi_\phi(\pi^*) + \log_2 e > 0.$$

Proposition 5.4 (Optimality Gap Lower-Bounds Accuracy)**.** *Under the conditions of the uniform convergence theorem, for any stochastic policy $\phi \in \mathcal{M}$, let*

$$\hat{\pi} \coloneqq \arg\max_{\pi \in \mathcal{A}^{\mathcal{S}}} \left\{ \alpha_{\text{train}}(\pi; \pi^*) - \sqrt{\frac{-2\chi_\phi(\pi) + \log_2 e - \log_2(1/\delta)}{2N}} \right\}, \tag{4}$$

then, with probability $1 - \delta$,

$$\alpha(\hat{\pi}; \pi^*) \geq 1 - \sqrt{\frac{2G(\phi; \pi^*) - 2\log_2(1/\delta)}{N}}. \tag{5}$$

Proposition 5.4 tells us that the goodness of a prior policy is decided by the coherence of the optimal d-policy under such a prior, where goodness is defined as the actual accuracy of the optimal d-policy when using such a prior for coherence regularization.

Note that both Definition 5.3 and Proposition 5.4 assume there is one fixed π^*, rather than a π^* drawn from a distribution $\mathcal{D} \in \Delta[\mathcal{A}^{\mathcal{S}}]$, as is the case in §3.1. However, Proposition 5.4 can be easily adapted to the latter

case, where instead of Equation 5, we have

$$\mathrm{E}_{\pi^* \sim \mathcal{D}}\left[\alpha(\hat{\pi}; \pi^*)\right] \geq 1 - \mathrm{E}_{\pi^* \sim \mathcal{D}}\left[\sqrt{\frac{2G(\phi; \pi^*) - 2\log_2(1/\delta)}{N}}\right].$$

5.2 Worst-Case Asymptotic Optimality of Coherence Regularization

The uniform convergence theorem tells us that coherence bounds the generalization gap. But which prior should we use? We now prove that the KL-optimal approximation to the data-generating distribution is the optimal choice.

Consider the semi-supervised learning setup from §3, with context partition $\mathcal{S}$, behavior space $\mathcal{A}$, d-policy space $\mathcal{A}^{\mathcal{S}}$, data-generating distribution $\mathcal{D} \in \Delta\left[\mathcal{A}^{\mathcal{S}}\right]$, context distribution $\mathcal{F} \in \Delta[\mathcal{S}]$, and N supervised samples. Under SRM with description-length regularization using prior $\mathcal{P}$, with

$$\pi_{\mathrm{SRM}} \coloneqq \underset{\pi \in \mathcal{A}^{\mathcal{S}}}{\arg\max}\ \alpha_{\mathrm{train}}(\pi) - \mathrm{reg}(\pi), \text{ where}$$
$$\mathrm{reg}(\pi) \coloneqq \left(\frac{-2\log_2 \mathcal{P}(\pi) + \log_2 e - \log_2(1/\delta)}{2N}\right)^{1/2},$$

we have the following optimality result.

Theorem 5.5 (Optimality of Description-Length Regularization)**.** *Under description-length regularization with prior $\mathcal{P}$, for any distribution $\mathcal{Q} \in \Delta[\mathcal{A}^{\mathcal{S}}]$ and sufficiently small δ such that $\log_2(1/\delta) \gg -\mathrm{E}_{\pi \sim \mathcal{Q}}[\log_2 \mathcal{P}(\pi)]$ (i.e., we consider worst-case outcomes at the lowest quantiles), with probability $1-\delta$,*

$$\alpha(\pi_{\mathrm{SRM}}) \geq \mathrm{E}_{\pi \sim \mathcal{Q}}[\alpha(\pi)] - \sqrt{\frac{2\log_2(1/\delta)}{N}} + \sqrt{\frac{2 + o(1)}{N\log_2(1/\delta)}}\ \left(\mathrm{H}\left[\mathcal{Q}\right] - \mathrm{KL}\left[\mathcal{Q} \,||\, \mathcal{P}\right]\right). \tag{6}$$

In particular, when $\mathcal{Q} = \mathcal{D}^{\beta}$ for $\beta > 0$,[a] the lower bound is maximized when $\mathcal{P}$ minimizes $\mathrm{KL}\left[\mathcal{D}^{\beta} \,||\, \mathcal{P}\right]$.

[a] $\mathcal{D}^{\beta}$ is the distribution obtained by raising $\mathcal{D}$'s probability masses to the β-th power before normalizing.

Proof. See Appendix D.4. □

Practically, let $\hat{\mathcal{D}}$ be the KL-optimal approximation to $\mathcal{D}$ learnable from the supervised samples, then, for $\mathcal{Q} = \mathcal{D}^{\beta}$, the optimal obtainable prior is $\mathcal{P} = \hat{\mathcal{D}}^{\beta}$, in line with Definition 3.8 when using the supervised policy as prior policy. This is why pretrained models are the natural choice of prior for coherence regularization. A pretrained language model, trained on the largest and most representative sample of $\mathcal{D}$, is the best available KL-approximation and therefore yields the tightest worst-case accuracy guarantee from Theorem 5.5.

The bound requires $\log_2(1/\delta) \gg -\log_2 \mathcal{P}(\pi)$, meaning we consider worst-case outcomes at the lowest quantiles (small δ). This is a strong assumption that may not hold when δ is not sufficiently small relative to the d-policy's description length. The next section examines the hypothesis that when the prior policy is trained on supervised samples, optimizing coherence alone is equivalent to optimizing the regularized training objective above.

5.3 Coherence with Trained Prior as Regularized Accuracy

The previous section established that coherence optimization is equivalent to description-length regularization. Proposition 5.4 tells us to optimize $\alpha_{\mathrm{train}}(\pi) - \mathrm{reg}(\pi)$, where $\mathrm{reg}(\pi)$ depends on coherence. But in practice (e.g., in Algorithm 1), we often optimize coherence alone, without an explicit training accuracy term. Why does this work?

The answer is that the prior policy already encodes the training samples. When we use a pretrained or supervised-trained model as the prior, optimizing coherence with respect to this prior implicitly optimizes

a joint objective that includes training accuracy. We hypothesize that the two formulations, (i) optimizing α_{train} + coherence regularization and (ii) optimizing coherence with a trained prior, are asymptotically equivalent under appropriate conditions.

To formalize this, we need machinery for reasoning about policies that have undergone two stages of training, namely pretraining and post-training.

Lemma 5.6 (Change-of-Prior)**.** *Denote*

$$\hat{\chi}_\phi[a_1, a_2, \cdots, a_k] \coloneqq \sum_{n=1}^{k} \log_2 \sigma\left(\phi + \sum_{m=1}^{n-1} a_m, s_n\right)(a_n),$$

where $a_n \in s_n$ for all n. For any policy $\rho \in \mathcal{M}, \phi = \rho + \sum_{n=1}^{N} a_n^\phi \in \mathcal{M}$ ($a_n^\phi \in \mathcal{A}$) and $\psi = \phi + \sum_{l=1}^{L} a_l^\psi \in \mathcal{M}$ ($a_l^\psi \in \mathcal{A}$), we have

$$\hat{\chi}_\phi[a_1^\psi, \cdots, a_L^\psi] + \hat{\chi}_\rho[a_1^\phi, \cdots, a_N^\phi] = \hat{\chi}_\rho[a_1^\phi, \cdots, a_N^\phi, a_1^\psi, \cdots, a_L^\psi].$$

Proof. See Appendix D.4. □

Definition 5.7 (Quotient System)**.** For a learning system $(\langle \mathcal{M}, + \rangle, \mathcal{A}, \mathcal{S}, \sigma)$, its *quotient system spanned by $\mathcal{S}_a \subseteq \mathcal{S}$ at $\phi \in \mathcal{M}$* is a learning system $(\langle \mathcal{M}_a, +_a \rangle, \mathcal{A}_a, \mathcal{S}_a, \sigma_a)$ defined by three properties. First, $\mathcal{A}_a = \{\phi + u : u \in \bigcup_{s \in \mathcal{S}_a} s\}$. Second, $\mathcal{M}_a = \langle \mathcal{A}_a \rangle_{+_a}$ with identity element $0_a \coloneqq \phi$, and the operator $+_a$ satisfying $(\phi + u) +_a (\phi + v) = \phi + (u + v)$. Third, $\sigma_a(\phi + u, s) = \sigma(\phi + u, s)$.

We denote such a relation with $(\langle \mathcal{M}_a, +_a \rangle, \mathcal{A}_a, \mathcal{S}_a, \sigma_a) \overset{\phi}{\trianglelefteq} (\langle \mathcal{M}, + \rangle, \mathcal{A}, \mathcal{S}, \sigma)$. We will denote with χ^a the coherence function in the quotient system.

Essentially, the quotient system spanned by $\mathcal{S}_a$ at ϕ represents the learning problem of post-training a pretrained policy ϕ on the context set $\mathcal{S}_a$. In the rest of this section, we will use "pretrain samples" to refer to our supervised samples, and "posttrain samples" to refer to our unsupervised samples, in line with the semi-supervised learning setup.

Theorem 5.8 (Prior Policy Encodes Previous Training Samples)**.** *Let*

$$(\mathcal{M}_a, \mathcal{A}_a, \mathcal{S}_a, \sigma_a) \overset{0_a}{\trianglelefteq} (\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$$
$$(\mathcal{M}_b, \mathcal{A}_b, \mathcal{S}_b, \sigma_b) \overset{0_b}{\trianglelefteq} (\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$$

such that $\mathcal{S}_a \cap \mathcal{S}_b = \emptyset$, $\mathcal{S}_a \cup \mathcal{S}_b = \mathcal{S}$, and, for some $\pi \in \mathcal{A}^{\mathcal{S}}$, $0_a = \sum_{s \in \mathcal{S}_b} \pi(s)$ and $0_b = \sum_{s \in \mathcal{S}_a} \pi(s)$. Then:

$$\underbrace{\chi^a(\pi(\mathcal{S}_a))}_{\substack{\text{posttrain coherence}\\ \text{w.r.t. pretrain prior}}} + \underbrace{\hat{\chi}_0\left[\pi(\mathcal{S}_b)\right]}_{\text{pretrain coherence}} = \chi(\pi) = \underbrace{\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]}_{\text{posttrain coherence}} + \underbrace{\chi^b(\pi(\mathcal{S}_b))}_{\substack{\text{posttrained accuracy}\\ \text{on pretrain samples}}}. \tag{7}$$

Proof. Both equalities follow from direct applications of the change-of-prior lemma. □

One way to interpret this theorem is that it shows the equivalence between:

LHS Optimizing solely for a policy's coherence on posttrain (unsupervised) contexts wrt pretrained (supervised) prior,[4] *and*

[4]The "pretrain coherence" term can be ignored as it's fixed.

RHS Optimizing jointly for the policy's coherence on posttrain (unsupervised) contexts and its accuracy on pretrain (supervised) samples.

Note that for the interpretation of RHS, there are still subtle gaps between the intuitive understanding here and the terms in equation 7, which we will revisit in Appendix D.5, in the heuristic argument for the conjecture below. We now conjecture the conditions under which optimizing coherence alone is equivalent to optimizing the regularized training objective.

Conjecture 5.9 (Equivalence of Coherence Optimization and Regularized Training)**.** *Let the training contexts π be randomly separated into two disjoint subsets, posttrain samples $\pi(\mathcal{S}_a)$ and pretrain samples $\pi(\mathcal{S}_b)$. The latter is given, while we need to optimize the former (which can be any d-policy in $\prod_{s\in\mathcal{S}_a} s$). I.e., we do* not *know the ground-truth labels for the posttrain contexts, and our task is to determine them.*

Consider (i) optimizing for coherence on the posttrain samples $\pi(\mathcal{S}_a)$ with respect to a pretrained policy or (ii) optimizing for $\alpha_{\text{train}}(\pi) - \text{reg}(\pi)$, i.e., SRM. The two are asymptotically equivalent when the posttrain sample count $|\mathcal{S}_a|$ is chosen to satisfy

$$\underbrace{|\mathcal{S}_a|}_{\text{posttrain sample count}} \sim \frac{1}{4} \times \underbrace{\frac{\overbrace{\left(-\chi^b(\pi^*(\mathcal{S}_b))/|\mathcal{S}_b|\right)^2}^{\text{mean pretrain coherence, squared}}}{-\hat{\chi}_0[\pi^*(\mathcal{S}_a)]/|\mathcal{S}_a|}}_{\text{mean posttrain coherence, inversed}} \times \underbrace{\left(\frac{1}{1-\alpha^b(\pi^*)}\right)^2}_{\substack{\text{pretrain error rate,}\\ \text{inverse-squared}}} \times \underbrace{|\mathcal{S}_b|}_{\text{pretrain sample count}} \tag{8}$$

See Appendix D.5 for a heuristic argument supporting the conjecture.

The conjecture states that two formulations are equivalent when the number of posttrain samples $|\mathcal{S}_a|$ is chosen appropriately. Even if we do not trust the exact estimate in Equation 8, we can use ternary search to find the optimal $|\mathcal{S}_a|$, since concavity of $f(r)$ and $g(r)$ implies that $\max_{r\in\mathbb{R}}(f(r) - \sqrt{|\mathcal{S}_a|}\cdot g(r))$ is unimodal in $|\mathcal{S}_a|$. In this sense, $|\mathcal{S}_a|$ is a hyperparameter whose optimal value we need to search for.

The equivalence explains why optimizing coherence alone works in practice. When we use a pretrained language model as the prior policy, we are implicitly optimizing for the same samples that prior policy was trained on. The prior encodes the training objective; optimizing coherence with respect to this prior recovers the regularized training objective.

The conjecture also suggests that the optimal number of posttrain samples scales with the effective number of pretrain samples that are i.i.d. with the posttrain contexts. This is likely much larger than typical in-context learning implementations (Xu et al., 2023; Wen et al., 2025), which already produce strong results. This motivates supervised training-based implementations of coherence optimization, which allow greatly expanding the number of unsupervised samples; we discuss the corresponding method in Appendix B.2.

Coherence optimization as transductive learning. Practical applications of coherence optimization are typically in-sample by design, similar to transductive learning. The goal is to find the best d-policy over a given set of contexts, and the "generalization" is from the supervised subset to the unsupervised subset of that same context set. Theorem 5.2 bounds accuracy on the full context set, which includes the unsupervised portion. In this sense, the unsupervised contexts serve simultaneously as the regularization medium and the evaluation target. This makes train/test splits less relevant than in standard supervised learning, because coherence optimization already evaluates each context relative to all others. Nevertheless, we also verify in Appendix E that coherence optimization improves accuracy on fully held-out test questions that were never part of the Gibbs sampling context.

Practical implications. The results in this section yield three actionable guidelines. First, the prior should be as close to the data-generating distribution as possible (Theorem 5.5). In practice, this means using a pretrained or supervised-trained model, and preferring priors that have been trained on data similar to the target task. Second, the number of unsupervised contexts for coherence optimization should be comparable

to the effective number of i.i.d. supervised samples (Conjecture 5.9). Too many unsupervised contexts can overwhelm the supervised signal; too few limit the regularization benefit. Third, coherence overoptimization is mitigable through early stopping, temperature tuning, or limiting the number of unsupervised contexts (Appendix 6.4).

6 Preliminary Experiments

We present preliminary experiments demonstrating that coherence is effective both as an evaluation metric and as a training objective. We also present practical results comparing types of prior policies, which agree with Theorem 5.5 on the data-vs-prior distributional distance being a determinant of effectiveness. Current results are preliminary and should be treated as a starting point for exploration.

6.1 Coherence as an Evaluation Metric

We test the scalability of coherence as an unsupervised proxy for truthfulness by using a 0.6B weak judge model/prior policy to scrutinize GPT-5 outputs twice, with classical LLM-as-a-Judge and with a coherence-based metric respectively.

Setup. Each data point represents a collection of around 5 HARDMath questions (Fan et al., 2024) with strong topical relevance. Blue points represent GPT-5-generated honest solutions to a collection, while red points represent GPT-5-generated deceptive solutions (where GPT-5 is explicitly prompted to generate "incorrect but convincing" solutions). In Figure 3(a) (LLM-as-a-Judge), the y-axis represents the sum of logprobs the Qwen3-0.6B weak judge model assigns to the solutions. In Figure 3(b) (coherence), the y-axis represents the difference between cross-solution coherence and sum of individual logprobs, i.e., pointwise mutual information (PMI), using the same weak judge as prior policy. The x-axis in both plots represents solution length.

Results. Note that this evaluation is inherently cross-question. Each solution's coherence is computed relative to other solutions in the same topical collection, so the metric already evaluates each output in the context of others rather than in isolation.

The coherence-based metric turns out much superior, as shown in Figure 3. If PMI (the difference between coherence and sum of logprobs) is positively predictive of honesty, it implies coherence provides additional predictive power over LLM-as-a-Judge. Indeed, deceptive solutions receive *higher* logprobs in LLM-as-a-Judge, meaning the judge is fooled by deception. They receive *lower* PMIs, meaning the coherence-based metric detects deception attempts better than LLM-as-a-Judge does. This demonstrates that cross-context internal coherence predicts truthfulness, even when the content is incomprehensible to the judge.

6.2 Coherence as an Optimization Objective

We test whether coherence optimization improves GSM8K performance for a 1B-parameter LLM by implementing Algorithm 2, across $\gamma \in \{0.25, 0.45, 0.65, 0.85\}$. We evaluate each setup over 100 random seeds and report the mean results.

Setup. We use `meta-llama/Llama-3.2-1B-Instruct` in an in-context learning (ICL) learning system and run $N = 40$ Gibbs rounds on a per-run training set $\mathcal{S}$ of size $|\mathcal{S}| = 100$. Accuracy is exact-match on the final numeric answer (parsed from the model output). We evaluate each configuration over 100 random seeds; for each seed we draw a fresh 100-example subset $\mathcal{S}$ from GSM8K and run the full γ sweep on this same subset, enabling paired comparisons across γ. Furthermore, we introduce a slightly modified sampling rule: for each $t \geq 1$ and each $s_n \notin \hat{\mathcal{S}}_t$, instead of sampling from $\sigma^\beta(\phi_t, s_n)$ we sample from an equal-weight mixture of the samplers induced by the base and current round contexts (ϕ_0, ϕ_t):

$$a_t^n \sim \tfrac{1}{2}\,\sigma^\beta(\phi_t, s_n) \;+\; \tfrac{1}{2}\,\sigma^\beta(\phi_0, s_n).$$

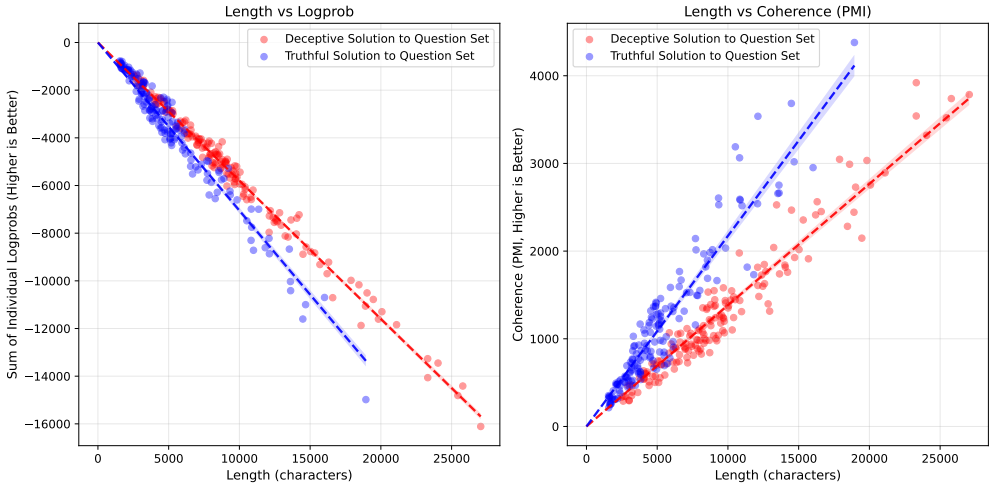

Figure 3: Coherence-based metrics detect deceptive solutions while LLM-as-a-Judge is fooled. Blue: honest solutions; Red: deceptive solutions. Left: LLM-as-a-Judge is fooled by deception and assigns higher probabilities (sum of logprobs) to deceptive solutions. Right: The coherence-based metric (PMI) successfully assigns higher scores to honest solutions.

Table 1: Aggregate summary statistics across Gibbs rounds. Mean ± SE across seeds.

γ	Rounds	Init Train (%)	Final Train (%)	Max Train (%)	Max Round	Improvement (%)
0.25	40	24.86 ± 0.48	$\mathbf{35.58 \pm 0.80}$	$\mathbf{45.90 \pm 0.62}$	19.6 ± 1.6	$\mathbf{+10.72 \pm 0.77}$
0.45	40	24.86 ± 0.48	33.56 ± 0.60	41.56 ± 0.51	19.1 ± 1.5	$+8.70 \pm 0.68$
0.65	40	24.86 ± 0.48	31.86 ± 0.64	38.40 ± 0.45	20.7 ± 1.5	$+7.00 \pm 0.66$
0.85	40	24.86 ± 0.48	27.72 ± 0.64	34.62 ± 0.55	20.4 ± 1.6	$+2.86 \pm 0.72$

In preliminary runs, sampling exclusively from the current-round context ϕ_t led to unstable dynamics: we observed occasional mode collapse and substantially higher variance across seeds. We theorize Gibbs updates can over-optimize toward a narrow region of high-coherence/low-performance π distributions, amplifying early-round noise and reducing exploration.

To mitigate this, we anchor sampling to the initial context ϕ_0 by drawing from an equal-weight mixture of $\sigma^\beta(\phi_t, \cdot)$ and $\sigma^\beta(\phi_0, \cdot)$. This can be viewed as a lightweight, training-free analogue of KL regularization. Empirically, this modification helped reduce mode collapse and yield more consistent trajectories while retaining performance across rounds.

Results. Figure 4 shows that coherence optimization consistently increases train accuracy over Gibbs rounds, with the strongest gains at smaller hold-out fractions. Across 100 paired seeds, $\gamma = 0.25$ achieves the best performance: mean train accuracy improves from 24.86% at initialization to 35.58% at the final round (+10.72%), with a peak of 45.90% on average (Table 1). As γ increases, both the final and peak accuracies monotonically decline (peak: 41.56% at $\gamma = 0.45$, 38.40% at $\gamma = 0.65$, and 34.62% at $\gamma = 0.85$), suggesting that resampling a larger complement (smaller γ) provides a stronger self-training signal in this learning system. Notably, the round of maximal performance is relatively stable across γ (roughly rounds 19-21 on average; Table 1), indicating that most gains occur in the first ∼20 rounds, after which improvement saturates. We hypothesize that this saturation arises from context-window limitations and inherent constraints of in-context learning (ICL), since the procedure can only improve predictions via contextual conditioning rather than explicit parameter updates.

While these results are in-sample, the observed saturation is consistent with a capacity ceiling: the `meta-llama/Llama-3.2-1B-Instruct` model reports 44.4% on GSM8K (Meta Llama, 2024). This is com-

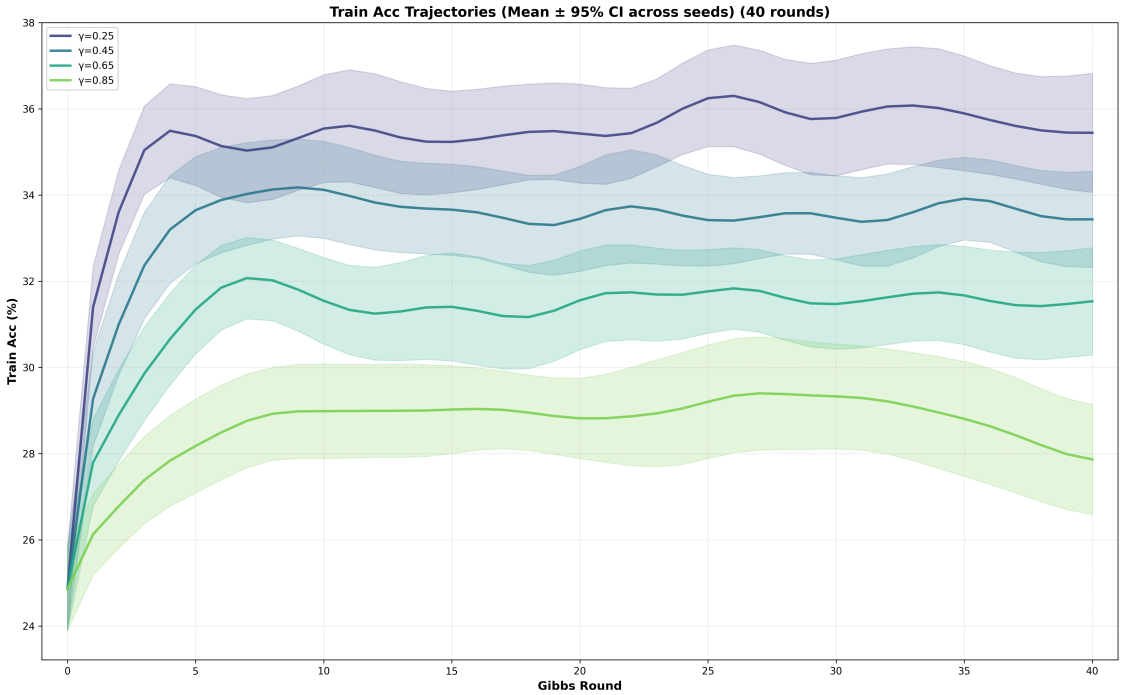

Figure 4: Mean train accuracy over 40 Gibbs rounds for $\gamma \in \{0.25, 0.45, 0.65, 0.85\}$ (shaded: 95% CI). Higher γ holds out a larger context subset $\hat{\mathcal{S}}_t$ when resampling.

parable in scale to our best average **in-sample** peak, but is not directly comparable due to differences in prompting/evaluation methods. **We report out-of-sample (test) accuracy on a separate set of** 200 **held-out GSM8K questions in Appendix E.** Test accuracy improves from roughly 30% to 40% under Gibbs sampling, confirming that coherence optimization generalizes beyond the in-sample context.

6.3 Effect of Prior Policy on Coherence Optimization

The choice of prior policy affects the performance of coherence optimization. We empirically examine how different types of priors influence the alignment between coherence and truthfulness.

Setup. We compare three types of prior policies. (1) **Pretrained prior** (BASE) is a base model trained only on next-token prediction. (2) **Posttrained prior** (CHAT) is a model further trained with RLHF for helpfulness and safety. (3) **Reinforced reasoning prior** (THINK) is a model trained with RL on verifiable reasoning tasks. We use the Qwen3 model family for all three classes of prior policies.

Under each prior policy, and for each of 8 models chosen from LMArena (Chiang et al., 2024), we compute the negated coherence of generations from the latter model on a self-curated set of open-ended research questions. Then, for each prior policy, we compute the Pearson's r between a model's LMArena Elo rating and its negated coherence under that prior policy (shown on y-axis, lower is better), and examines how it changes w.r.t. the prior policy's parameter count (shown on x-axis). This would indicate the extent to which scaling is possible for coherence optimization, and the comparison of practical fit across types of prior policy.

Results. Figure 5 shows accuracy scaling trends for each prior type. Several patterns emerge.

When the model is small, all three prior types perform similarly, encoding basic grammar, logic, and commonsense. As model size increases, we observe different scaling slopes. A negative slope indicates that higher-fidelity modeling of that prior leads to better coherence-truthfulness alignment. We observe

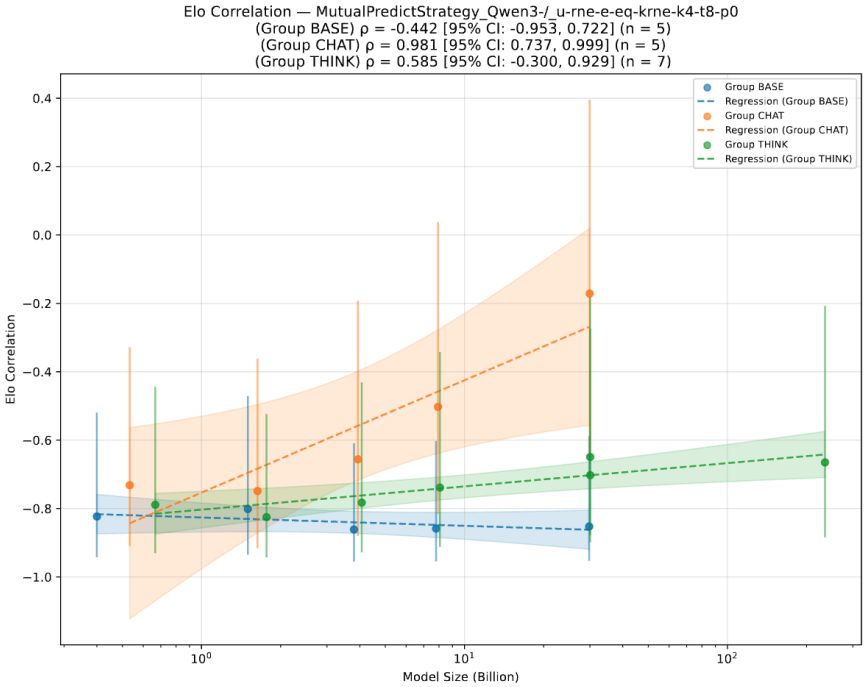

Figure 5: Scaling trends of coherence-truthfulness agreement for different prior policies in an in-context learning system. Lower is better. BASE refers to a pretrained prior, CHAT to a posttrained/RLHF prior, and THINK to a reinforced reasoning prior.

$\text{slope}_{\text{BASE}} < \text{slope}_{\text{THINK}} < \text{slope}_{\text{CHAT}}$. This suggests that adding the preference prior (CHAT) de-aligns coherence from accuracy, while further adding the factuality prior (THINK) re-aligns them.

The pretrained and reinforced reasoning priors follow the strategy of "using all available supervision to train a best policy, then let coherence optimization regularize with that policy as prior." This is optimal in the sense that these priors best approximate the true data-generating distribution among the options available.

6.4 Discussion

These experiments, while preliminary, reveal several patterns worth examining.

Coherence Detects Deception When Direct Evaluation Fails. The HARDMath results show that a weak judge's coherence-based metric outperforms its direct evaluation, even though both use the same model. How can comparing multiple outputs reveal information that evaluating a single output cannot?

A deceptive solution must simultaneously appear plausible to the judge and contain the deceptive content. The first condition is easy to satisfy in isolation, but the second introduces constraints that conflict with the model's learned distribution over related content. A deceptive mathematical argument must avoid certain true statements that would expose the deception. This avoidance makes the deceptive solution locally plausible but globally improbable when considered alongside topically related solutions. PMI captures this. High marginal probability but low joint probability indicates a solution optimized to appear good in isolation while avoiding broader context that would expose it.

An adversary can optimize for marginal plausibility, but optimizing for joint plausibility across many related contexts is equivalent to actually being truthful.

Note, however, the HARDMath experiment used synthetic deceptive solutions generated by prompting; it remains unclear whether the same signal would detect naturally occurring or adversarially trained deception.

Coherence Overoptimization (Goodharting). In practice, we sometimes see response texts becoming unnatural from the human perspective (yet still information-rich and score highly on automated benchmarks), after many rounds of Gibbs sampling. This may be due to limitations of in-context learning, where the objective is not exactly to imitate in-context examples, or it may be due to *overoptimization* of coherence. Below we briefly summarize the types of coherence overoptimization that are theoretically possible.

1. **Small divergence between desired measurement target (e.g. human legibility) and actual coherence measurement (e.g. pretrained policy).**

   Its impact on accuracy is modest and can be reflected by the increase in the optimality gap $G(\phi; \pi^*) = -2\chi_\phi(\pi^*) + \log_2 e$, whose continuity ensures that a small divergence only leads to small change in the gap size.

   However, its impact on alignment can be arbitrarily large, assuming that alignment is carried out by instilling a certain bias (e.g. human legibility) in the prior policy. As we have seen in Gao et al. (2023), optimizing for a flawed proxy for alignment bias can lead to catastrophic deviation from the desired bias (e.g. complete loss of human legibility).

2. **Too many coherence optimization contexts.** The informal theorem on joint optimization has determined the optimal number of contexts for coherence optimization to be on the same order as the effective number of i.i.d. supervised training samples. While such a number is hard to exceed in practice, when the coherence optimization contexts focus on a certain niche in the context space, they can easily outnumber the effective supervised samples in the same niche.

   Its impact on accuracy can be arbitrarily large. For instance, when the coherence optimization contexts are entirely orthogonal to supervised training samples, or only represent a vanishingly small portion of those samples, the optimality gap $G(\phi; \pi^*) = -2\chi_\phi(\pi^*) + \log_2 e$ goes to infinity.

   It has no negative impact on alignment, since alignment is itself embodied by the coherence optimization process, rather than being traded off against.

In both cases, to mitigate overoptimization, we may use early-stopping on three different aspects.

1. **Early stopping on Gibbs iterations**, i.e., to stop Gibbs sampling after a certain number of rounds when we see that desirable properties (e.g. accuracy, human legibility) start to decline.
2. **Reducing number of coherence optimization contexts**, typically to the same ballpark as the effective number of supervised learning samples.
3. **Higher temperature in the "softmax over coherence" distribution**, which in the Gibbs sampling case, implies a reduced inverse temperature β.

For all three interventions, we can use ternary search or similar methods to find the sweet spot, assuming the ability to evaluate desired properties (e.g. accuracy, human legibility).

7 Conclusion

This paper presented coherence optimization as a unified framework for feedback-free self-improvement. We showed that state-of-the-art methods (debate, bootstrap, ICM) are instances of coherence optimization, proved that coherence regularization is optimal among description-length regularizers, developed a principled method for coherence optimization, and demonstrated preliminary empirical evidence for its effectiveness.

Appendix A addresses a list of frequently asked questions.

Limitations and Future Work

One major limitation concerns experimental validation. Existing experiments in §6 test key elements in the theory such as coherence-truthfulness alignment, prior choice, and accuracy uplift, but do not directly

validate the claim of optimality by comparing against alternative regularizers. A rigorous comparison would require matching compute budgets across coherence regularization, entropy regularization, KL-divergence to a reference policy, and consistency regularization on a common semi-supervised task. Such a comparison is an important direction for future work.

Additionally, our analysis suggests larger potential for coherence optimization if scaled to near-pretrain scale. By allowing a large reasoning model to perform extended reasoning on *all* the contexts in its pretraining data, it can discover the belief system that most coherently explains the world. This kind of coherence-seeking across otherwise unrelated contexts has given rise to many of humanity's most important realizations, e.g., special relativity arose from the inconsistency between the fixed speed of light and the behavior of fast-moving objects, and the immorality of slavery was recognized through the inconsistency between the institution and principles of innate human rights. We have also shown that two-agent debate is a special case of coherence optimization, and such debates occur in almost every instance of group deliberation. Coherence optimization thus serves as a principled model of reflection in both humans and AIs, and future work may explore its extension to multi-agent and human-AI systems.

Broader Impact Statement

This work proposes coherence optimization as a framework for feedback-free self-improvement in language models. While the primary motivation is to improve model truthfulness without requiring external supervision, we acknowledge potential dual-use concerns. On the positive side, coherence optimization could enhance the reliability of AI systems by identifying and resolving internal inconsistencies, and develops a characterization of what self-improvement methods truly optimize for, both important for AI safety. On the negative side, any method that improves model capabilities could potentially be misused or contribute to loss-of-control risks. We believe the benefits of developing mechanistic understanding of unsupervised self-improvement outweigh risks brought by the capablity uplifts already mostly exhausted by existing methods.

Acknowledgments

Almost all substantial content is manually drafted, with editorial assistance from AI. Standard results, examples, straightforward explanation of setups, the illustrative Figure 2, and connective content are AI-drafted with manually-written outline, and undergoes extensive manual review and editing.

References

Eyar Azar and Boaz Nadler. Semi-supervised sparse gaussian classification: Provable benefits of unlabeled data. *Advances in Neural Information Processing Systems*, 37:20132–20169, 2024.

Bert Baumgaertner and Charles Lassiter. Precedent and rest stop convergence in reflective equilibrium. *Synthese*, 203(3):88, 2024.

Claus Beisbart. Making reflective equlibrium precise: A formal model. *Ergo an Open Access Journal of Philosophy*, 2021.

Shai Ben-David, Tyler Lu, and Dávid Pál. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *Conference on Learning Theory (COLT)*, pp. 33–44, 2008.

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022.

Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. Scalable ai safety via doubly-efficient debate. *International Conference on Machine Learning (ICML 2024)*, 2024.

Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. Avoiding obfuscation with prover-estimator debate. *arXiv preprint arXiv:2506.13609*, 2025.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. Self-icl: Zero-shot in-context learning with self-generated demonstrations. *arXiv preprint arXiv:2305.15035*, 2023.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.

Norman Daniels. *Justice and justification: Reflective equilibrium in theory and practice*. Cambridge University Press, 1996.

Finnur Dellsén. Probabilifying reflective equilibrium. *Synthese*, 203(2):45, 2024.

Michael R DePaul. *Balance and refinement: Beyond coherence methods of moral inquiry*. Routledge, 2006.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

Jingxuan Fan, Sarah Martinson, Erik Y Wang, Kaylie Hausknecht, Jonah Brenner, Danxian Liu, Nianli Peng, Corey Wang, and Michael P Brenner. Hardmath: A benchmark dataset for challenging problems in applied mathematics. *arXiv preprint arXiv:2410.09988*, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5000–5011, 2021.

DW Haslett. What is wrong with reflective equilibria? *The Philosophical Quarterly (1950-)*, 37(148): 305–311, 1987.

Junlin He, Jinxiao Du, and Wei Ma. Preventing dimensional collapse in self-supervised learning via orthogonality regularization. In *Advances in Neural Information Processing Systems*, 2024.

Margaret Holmgren. The wide and narrow of reflective equilibrium. *Canadian Journal of Philosophy*, 19(1): 43–60, 1989.

Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Jiayi Zhou, Borong Zhang, Donghai Hong, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, et al. Ai alignment: A contemporary survey. *ACM Computing Surveys*, 2025.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*, 2025.

Zachary Kenton, Noah Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah Goodman, et al. On scalable oversight with weak llms judging strong llms. *Advances in Neural Information Processing Systems*, 37:75229–75276, 2024.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive llms leads to more truthful answers. *arXiv preprint arXiv:2402.06782*, 2024.

Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025.

Hunter Lang, David Sontag, and Aravindan Vijayaraghavan. Theoretical analysis of weak-to-strong generalization. *Advances in neural information processing systems*, 37:46837–46880, 2024.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Jan Leike. *Nonparametric general reinforcement learning*. PhD thesis, Australian National University, 2016.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch. Composing ensembles of pre-trained models via iterative consensus. *arXiv preprint arXiv:2210.11522*, 2022.

Richard Lohse. Overlapping consensus in pluralist societies: simulating rawlsian full reflective equilibrium. *Synthese*, 203(1):11, 2023.

Yury Maximov, Massih-Reza Amini, and Zaid Harchaoui. Rademacher complexity bounds for a penalized multi-class semi-supervised algorithm. *Journal of Artificial Intelligence Research*, 61:761–786, 2018.

Meta Llama. Llama-3.2-1b-instruct model card, 2024. `https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct`.

Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, and Ard A Louis. Is sgd a bayesian sampler? well, almost. *Journal of Machine Learning Research*, 22(79):1–64, 2021.

John Rawls. A theory of justice. *The Law of Peoples*, 67, 1971.

John Rawls. *Political Liberalism*. Columbia University Press, 2005.

Utkarsh Sharma and Jared Kaplan. Scaling laws from the data manifold dimension. *Journal of Machine Learning Research*, 23(9):1–34, 2022.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Ray J Solomonoff. Algorithmic probability: Theory and applications. In *Information theory and statistical learning*, pp. 1–23. Springer, 2009.

Folke Tersman. Seeking a reflective equilibrium in the face of disagreement. *Synthese*, 204(3):86, 2024.

Einar Urdshals, Edmund Lau, Jesse Hoogland, Stan van Wingerden, and Daniel Murfet. Compressibility measures complexity: Minimum description length meets singular learning theory. *arXiv preprint arXiv:2510.12077*, 2025.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10 (5):988–999, 1999.

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2021.

Jiaxin Wen, Zachary Ankner, Arushi Somani, Peter Hase, Samuel Marks, Jacob Goldman-Wetzler, Linda Petrini, Henry Sleight, Collin Burns, He He, et al. Unsupervised elicitation of language models. *arXiv preprint arXiv:2506.10139*, 2025.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Weijia Xu, Andrzej Banburski-Fahey, and Nebojsa Jojic. Reprompting: Automated chain-of-thought prompt inference through gibbs sampling. *arXiv preprint arXiv:2305.09993*, 2023.

A Frequently Asked Questions

In this section, we address potential questions about the overall framework.

Does coherence optimization just find any "coherent-but-false" peak, e.g., conspiracy theories?

Critique: Coherence optimization will find any peak of high internal consistency, even one that is factually wrong. It can't distinguish a "true" equilibrium from a "coherent-but-false" one.

No. Coherence optimization does not happen in a vacuum. It is a *regularization technique* that functions on top of supervised signals, as established in the theoretical results (e.g., Proposition 5.4). Although the coherence optimization process itself uses no additional labels, the supervised signal enters through the pretrained prior, which encodes the knowledge from its training data. This is consistent with the semi-supervised framing in the introduction and §3.1.

- In Language Models: Coherence optimization inherits the vast knowledge embedded in the pretrained model. This prior *already* constrains the search space, making most "coherent-but-false" peaks (which contradict pretraining data) have a much lower coherence score from the start. Also, this distinguishability scales with reasoning strength; as we train stronger and stronger reasoner models, their ability to reason at length enhances immunity against hard-to-find incoherence that would not be obvious from just looking at the pretraining data.
- In Reinforcement Learning Agents: The agent receives Bernoulli feedback from the environment, which mixes both subjective coherence and the "supervised signal" to ground the coherence search.

It remains an open question whether these signals are sufficient to *uniquely* determine the globally optimal reflective equilibrium. However, they immensely narrow the search space. Historically, ideologies that are wrong in hindsight (e.g., the moral defense of slavery) are almost always revealed through their *incoherence against new, incoming observations* (e.g., the philosophical idea of universal human rights). Coherence optimization is the formal process for identifying and resolving exactly this type of inconsistency.

Can we customize coherence to optimize for properties other than truthfulness?

Critique: The paper focuses on truthfulness, but what if we care about other properties like human-readability or value alignment?

Yes. The prior policy ϕ can be chosen to regularize for different traits. Recall that coherence $\chi_\phi(\pi)$ measures the log-probability of d-policy π under prior ϕ. By choosing ϕ appropriately, we can steer the optimization toward different goals:

- **Truthfulness:** Use a pretrained model as ϕ. The pretrained model has seen vast amounts of factual data, so d-policies that contradict well-established facts will have low coherence.
- **Human legibility:** Use a model that predicts what humans find easy to understand as ϕ. Since negated coherence equals description length, maximizing coherence under a "human prior" means minimizing the description length of the policy from a human perspective. This incentivizes behaviors that are predictable and interpretable to humans.
- **Value alignment:** Use an aligned model (e.g., one trained with RLHF on safety-focused data) as ϕ. D-policies exhibiting misaligned behaviors, such as deception or sandbagging, will incur high description length under an aligned prior, since such behaviors are hard for an aligned model to predict.

In principle, any trainable model can serve as a prior, including human preference models. The role of coherence optimization is to turn such a prior (which evaluates individual outputs) into a policy-level regularizer (which evaluates the joint distribution of outputs across contexts).

Why regularize at the policy level instead of per-output?

Critique: Standard methods like RLHF already regularize model outputs. What's the advantage of policy-level regularization?

RLHF and similar methods apply supervision to the *marginal distribution*, rewarding or penalizing individual outputs in isolation. Policy-level regularization, by contrast, considers the *joint distribution* of outputs across all contexts.

This distinction matters because certain failure modes are invisible in marginals but obvious in joints. Figure 6 illustrates the problem. Given only marginal distributions (the "shadow" of each peak), you cannot tell which peak is tallest. But the joint distribution (the full 3D landscape) makes the answer obvious. Consider sandbagging, where a model performs poorly on some tasks but well on others. Each individual output might look reasonable in isolation; the problem only becomes apparent when comparing outputs across contexts. Deception works similarly. A lie in one context might seem plausible until you notice it contradicts what the model said elsewhere.

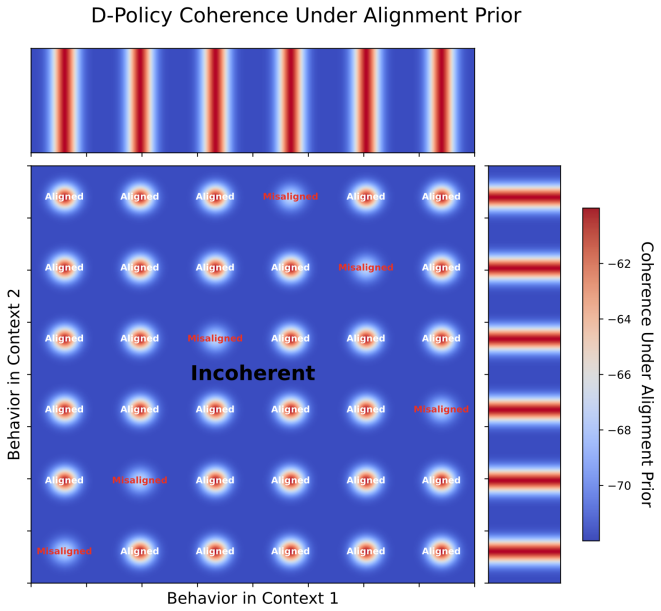

Figure 6: Illustration of d-policy space. Peaks represent coherent d-policies. Given only marginal distributions (the "shadow" of each peak), it is impossible to distinguish the tall peak from the short ones. Coherence optimization selects for high joint probability, naturally steering toward the tallest peak.

Coherence optimization naturally catches these failures. A d-policy that sandbags or deceives will have low coherence, because such behaviors are hard to predict from the model's other outputs. The inconsistency manifests as high description length.

Does coherence optimization require a fully truthful model to begin with?

Critique: If we need a truthful prior to improve truthfulness, don't we already need to solve the problem first?

Not necessarily. What we need is that the prior policy be *substantially truthful* in a precise sense. The optimality gap $G(\phi; \pi^*) = -2\chi_\phi(\pi^*) + \log_2 e$ of the ground-truth d-policy π^* must be smaller than the expected optimality gap $\mathrm{E}_{\pi'}[G(\phi; \pi')]$ for a random d-policy π' drawn from the prior. In other words, truth must outperform the average response under the prior, though it need not be the most likely answer to every individual question. The prior is allowed to sometimes produce falsehoods, and coherence optimization corrects such errors by favoring the globally most coherent d-policy. When this condition is violated, the generalization bound (Proposition 5.4) becomes vacuous and coherence optimization may converge to a

coherent-but-false equilibrium. This is exactly the failure mode the theory predicts. As practical strategies for ensuring this condition holds, we may consider the following.

1. **Interleaved training:** Start from a base pretrained model. Use it as the prior while simultaneously doing coherence optimization and capability training. The pretrained prior constrains the training to stay consistent with the factual knowledge already encoded.
2. **Weak-to-strong:** Use a smaller, well-understood model as the prior for training a larger model. The smaller model's factual grounding transfers to the larger model through coherence regularization, even though the smaller model is less capable.

The circularity is broken because coherence optimization amplifies this existing truthful tendency in pretrained models rather than creating it from scratch.

B Examples and Proofs for Gibbs Sampling-Based Coherence Optimization

This section provides detailed examples and formal proofs for the Gibbs sampling-based coherence optimization methods.

B.1 The Original Gibbs Sampling Method

Example B.1 (Gibbs Sampling Over Sauces)**.** Let us trace one full update step of the algorithm, starting from a plausible but non-optimal d-policy $\pi_0 = (a^{\text{burger}}_{\text{mustard}}, a^{\text{fries}}_{\text{ketchup}})$. Let the temperature reciprocal $\beta = 1$ (i.e., no temperature scaling).

Sample the new burger behavior, a^1_0. The context for this sample is the *other* component of the current d-policy, which is $\pi_0(s^{\text{fries}}) = a^{\text{fries}}_{\text{ketchup}}$. We sample from the distribution $\sigma(a^{\text{fries}}_{\text{ketchup}}, s^{\text{burger}})$, which in this Bayesian system is the conditional probability distribution $P(s^{\text{burger}}|a^{\text{fries}}_{\text{ketchup}})$. Using the prior table (with $\epsilon = 0$): First, find the marginal probability of the context: $P(a^{\text{fries}}_{\text{ketchup}}) = P(a^{\text{burger}}_{\text{mayo}}, a^{\text{fries}}_{\text{ketchup}}) + P(a^{\text{burger}}_{\text{mustard}}, a^{\text{fries}}_{\text{ketchup}}) + P(a^{\text{burger}}_{\text{other}}, a^{\text{fries}}_{\text{ketchup}}) = 0 + 0.175 + 0.175 = 0.35$. Next, compute the conditional probabilities for each burger choice: $P(a^{\text{burger}}_{\text{mayo}}|a^{\text{fries}}_{\text{ketchup}}) = P(a^{\text{burger}}_{\text{mayo}}, a^{\text{fries}}_{\text{ketchup}})/P(a^{\text{fries}}_{\text{ketchup}}) = 0/0.35 = 0$; $P(a^{\text{burger}}_{\text{mustard}}|a^{\text{fries}}_{\text{ketchup}}) = P(a^{\text{burger}}_{\text{mustard}}, a^{\text{fries}}_{\text{ketchup}})/P(a^{\text{fries}}_{\text{ketchup}}) = 0.175/0.35 = 0.5$; and $P(a^{\text{burger}}_{\text{other}}|a^{\text{fries}}_{\text{ketchup}}) = P(a^{\text{burger}}_{\text{other}}, a^{\text{fries}}_{\text{ketchup}})/P(a^{\text{fries}}_{\text{ketchup}}) = 0.175/0.35 = 0.5$. We then sample a^1_0 from this distribution: $\{\text{mayo: } 0, \text{mustard: } 0.5, \text{other: } 0.5\}$. Suppose we sample $a^1_0 = a^{\text{burger}}_{\text{mustard}}$.

Sample the new fries behavior, a^2_0. The context for this sample is $\pi_0(s^{\text{burger}}) = a^{\text{burger}}_{\text{mustard}}$. We sample from the distribution $\sigma(a^{\text{burger}}_{\text{mustard}}, s^{\text{fries}})$, which is $P(s^{\text{fries}}|a^{\text{burger}}_{\text{mustard}})$. The marginal probability of the context is $P(a^{\text{burger}}_{\text{mustard}}) = 0 + 0.175 + 0.175 = 0.35$. The conditional probabilities for each fries choice are: $P(a^{\text{fries}}_{\text{mayo}}|a^{\text{burger}}_{\text{mustard}}) = 0/0.35 = 0$; $P(a^{\text{fries}}_{\text{ketchup}}|a^{\text{burger}}_{\text{mustard}}) = 0.175/0.35 = 0.5$; and $P(a^{\text{fries}}_{\text{other}}|a^{\text{burger}}_{\text{mustard}}) = 0.175/0.35 = 0.5$. We then sample a^2_0 from $\{\text{mayo: } 0, \text{ketchup: } 0.5, \text{other: } 0.5\}$. We sample $a^2_0 = a^{\text{fries}}_{\text{other}}$.

Construct the new d-policy. The new d-policy is $\pi_1 = (a^1_0, a^2_0) = (a^{\text{burger}}_{\text{mustard}}, a^{\text{fries}}_{\text{other}})$. The system has transitioned to a new d-policy, which has the same coherence as the initial one: $\chi(\pi_1) = \log_2(0.175) \approx -2.51$.
The run has not converged yet, but we stop here for illustration. Notice that the most coherent d-policy $\pi^*_1 = (a^{\text{burger}}_{\text{mayo}}, a^{\text{fries}}_{\text{mayo}})$ is an absorbing state. If we ever sample $a^{\text{burger}}_{\text{mayo}}$, the next sample for fries will be $a^{\text{fries}}_{\text{mayo}}$ with probability 1, as $P(a^{\text{fries}}_{\text{mayo}}|a^{\text{burger}}_{\text{mayo}}) = 1$. Symmetrically, if we sample $a^{\text{fries}}_{\text{mayo}}$, the next burger choice must be $a^{\text{burger}}_{\text{mayo}}$. The sampler cannot escape this state, which has the highest coherence. For an ergodic system (e.g., if $\epsilon > 0$), the sampler would not get stuck but would instead visit this high-coherence state more frequently than others, eventually sampling from the softmax distribution over all d-policies.

We see that Gibbs sampling tends to converge upon high-coherence d-policies. We now prove that Bayesian learning systems with nonzero priors satisfy the ergodicity condition (Definition 4.1).

Example B.2 (Bayesian Learning Systems With Nonzero Priors Are Ergodic)**.** Recall that a system is ergodic if for any two d-policies π, π' that differ in only one context $\dot{s}$, the transition probability $\sigma(\sum_{s \neq \dot{s}} \pi(s), \dot{s})(\pi'(\dot{s}))$ is greater than zero.

In a Bayesian learning system, this transition probability is the conditional probability $P(\pi'(\dot{s}) | \{\pi(s)\}_{s \neq \dot{s}})$. By the definition of conditional probability,

$$P(\pi'(\dot{s}) | \{\pi(s)\}_{s \neq \dot{s}}) = \frac{P(\pi'(\dot{s}), \{\pi(s)\}_{s \neq \dot{s}})}{P(\{\pi(s)\}_{s \neq \dot{s}})}.$$

The d-policy $(\pi'(\dot{s}), \{\pi(s)\}_{s \neq \dot{s}})$ is simply π'. A Bayesian system with a nonzero prior assumes that the joint probability $P(\pi_{joint})$ is strictly positive for *any* full d-policy π_{joint}.

Therefore, the numerator $P(\pi')$ is greater than zero. The denominator $P(\{\pi(s)\}_{s \neq \dot{s}})$ is a marginal probability, calculated by summing the joint probabilities over all possible behaviors in $\dot{s}$:

$$P(\{\pi(s)\}_{s \neq \dot{s}}) = \sum_{a \in \dot{s}} P(a, \{\pi(s)\}_{s \neq \dot{s}}).$$

Since every term in this sum is strictly positive by assumption, the denominator is also strictly positive. Thus, the conditional probability is the ratio of two positive numbers, which is positive. This holds for any pair of d-policies differing by a single component, so the system is ergodic.

Theorem B.3 (Gibbs Sampling Recovers Softmax Over Coherence)**.** *For any ergodic learning system* $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$*, initial d-policy* π_0*, and temperature reciprocal* $\beta > 0$*, when* $N \to +\infty$*, we have*

$$\pi_r \xrightarrow{d} \mathrm{X}^\beta, \quad \textit{where } r \sim \mathrm{Unif}(\{0, \cdots, N\})$$

Proof. The Gibbs sampling algorithm defines a time-homogeneous Markov chain on the finite state space of d-policies $\mathcal{A}^{\mathcal{S}}$. To show that the distribution of states converges to X^β, we need to show that the chain is ergodic and that X^β is its unique stationary distribution.

Ergodicity: The chain is irreducible because the learning system is assumed to be ergodic. By definition, this means for any π, π' differing by one component, the transition probability is positive. Since any d-policy can be reached from any other d-policy by a sequence of single-component changes, there is a non-zero probability path between any two states in the chain. The chain is also aperiodic; for example, there is a non-zero probability of resampling the same behavior for a component, allowing the chain to stay in the same state. An irreducible and aperiodic Markov chain on a finite state space is ergodic.

Stationary Distribution: We show that X^β satisfies the detailed balance condition, which is a sufficient condition for being a stationary distribution. For any two states $\pi, \pi' \in \mathcal{A}^{\mathcal{S}}$, the condition is $\mathrm{X}^\beta(\pi) P(\pi \to \pi') = \mathrm{X}^\beta(\pi') P(\pi' \to \pi)$. It suffices to check this for states π, π' that can be reached from one another in a single step, i.e., those that differ in at most one component. If $\pi = \pi'$, the condition is trivial. Let π and π' differ only in a single context, $\dot{s}$. Let $\pi(\dot{s}) = a$ and $\pi'(\dot{s}) = a'$. Let $\phi = \sum_{s \neq \dot{s}} \pi(s) = \sum_{s \neq \dot{s}} \pi'(s)$. The transition probability $P(\pi \to \pi')$ involves updating the component $\dot{s}$ to a'. The probability for this specific update is proportional to $\sigma^\beta(\phi, \dot{s})(a')$. We write the full transition probability, accounting for the multi-step nature of the update: $P(\pi \to \pi') = \frac{1}{|\mathcal{S}|} \frac{\sigma(\phi, \dot{s})(a')^\beta}{Z_\phi}$, where $Z_\phi = \sum_{\tilde{a} \in \dot{s}} \sigma(\phi, \dot{s})(\tilde{a})^\beta$ is the normalization constant.

The key insight is that the coherence $\chi(\pi)$ is independent of the ordering of contexts. We can therefore choose an ordering where $\dot{s}$ is the last element, $s_{|\mathcal{S}|}$. With this ordering, the coherence of π is:

$$\chi(\pi) = \left(\sum_{n=1}^{|\mathcal{S}|-1} \log_2 \sigma \left(\sum_{m=1}^{n-1} \pi(s_m), s_n \right) (\pi(s_n)) \right) + \log_2 \sigma(\phi, \dot{s})(a)$$

And for π':

$$\chi(\pi') = \left(\sum_{n=1}^{|\mathcal{S}|-1} \log_2 \sigma \left(\sum_{m=1}^{n-1} \pi(s_m), s_n \right) (\pi(s_n)) \right) + \log_2 \sigma(\phi, \dot{s})(a')$$

The terms in the parentheses are identical for π and π'. Therefore,

$$\chi(\pi) - \chi(\pi') = \log_2 \sigma(\phi, \dot{s})(a) - \log_2 \sigma(\phi, \dot{s})(a')$$

Taking the exponential and raising to the power of β:

$$\frac{2^{\beta\chi(\pi)}}{2^{\beta\chi(\pi')}} = \frac{\sigma(\phi, \dot{s})(a)^\beta}{\sigma(\phi, \dot{s})(a')^\beta}$$

Now, let's check the detailed balance condition using $\mathrm{X}^\beta(\pi) \propto 2^{\beta\chi(\pi)}$:

$$\mathrm{X}^\beta(\pi) P(\pi \to \pi') \propto 2^{\beta\chi(\pi)} \cdot \sigma^\beta(\phi, \dot{s})(a') = 2^{\beta\chi(\pi)} \cdot \frac{\sigma(\phi, \dot{s})(a')^\beta}{Z_\phi}$$

$$\mathrm{X}^\beta(\pi') P(\pi' \to \pi) \propto 2^{\beta\chi(\pi')} \cdot \sigma^\beta(\phi, \dot{s})(a) = 2^{\beta\chi(\pi')} \cdot \frac{\sigma(\phi, \dot{s})(a)^\beta}{Z_\phi}$$

The condition holds if $2^{\beta\chi(\pi)}\sigma(\phi, \dot{s})(a')^\beta = 2^{\beta\chi(\pi')}\sigma(\phi, \dot{s})(a)^\beta$, which is equivalent to $\frac{2^{\beta\chi(\pi)}}{2^{\beta\chi(\pi')}} = \frac{\sigma(\phi,\dot{s})(a)^\beta}{\sigma(\phi,\dot{s})(a')^\beta}$. We have just shown this identity to be true.

Thus, X^β is the stationary distribution. Since the chain is ergodic, it has a unique stationary distribution, so the time-averaged distribution of samples π_r converges to X^β. □

By running Gibbs sampling with large β, we are now able to find high-coherence d-policies.

B.2 The Training-Friendly Variant

In practice, Gibbs sampling is easily applicable on in-context learning systems and Bayesian inference systems. However, each of its $N \cdot |\mathcal{S}|$ sampling steps samples from a different policy (obtained from leave-one-out datasets), and in a finetuning learning system, it would be a computational nightmare to do this many training runs.

If one is fine with approximations, then they could train one full policy $\sum_{n=1}^{|\mathcal{S}|} \pi_t(s_n)$ in each round, and before every sampling attempt, perform **unlearning** to erase one sample from such a policy.

Or, if the policy at hand can be easily merged (e.g. are ensemble models), then one may precompute prefix and suffix "sum policies", and merge one prefix and one suffix to obtain a leave-one-out policy. If storage is a concern, one could further apply square root decomposition.

However, the two workarounds are rather complicated and may not be applicable in many cases. Below is a more general method.

Algorithm 2 Training-Friendly Variant of Gibbs Sampling

Input: Learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$; initial d-policy π_0; number of steps N; temperature reciprocal β; **training split portion** $\gamma \in (0, 1)$.
Output: List of d-policies $\pi_0, \pi_1, \cdots, \pi_N$.
procedure TrainingFriendlyGibbs
  for $t = 0, 1, \ldots, N-1$ **do**
    Independently and equiprobably select a $\lfloor \gamma|\mathcal{S}| \rfloor$-sized subset of $\mathcal{S}$, denoted with $\hat{\mathcal{S}}_t$.
    Let $\phi_t = \sum_{s \in \hat{\mathcal{S}}_t} \pi_t(s)$. **(Training step)**
    Independently sample $a_t^n \sim \sigma^\beta(\phi_t, s_n)$, for $s_n \notin \hat{\mathcal{S}}_t$. **(Sampling step)**
    Set π_{t+1} such that $\pi_{t+1}(s_n) = a_t^n$ if $s_n \notin \hat{\mathcal{S}}_t$ and $\pi_{t+1}(s_n) = \pi_t(s_n)$ if $s_n \in \hat{\mathcal{S}}_t$.
  end for
end procedure

This algorithm is first implemented in an forthcoming, unpublished version of Wen et al. (2025). Note that it needs to be distinguished from what is typically called "iterative training", as the training step in each round always starts from zero, rather than from the previous step's trained policy ϕ_{t-1}. Iterative training usually results in "model collapse" (Shumailov et al., 2024).

This algorithm approximates Gibbs sampling by replacing the computationally expensive leave-one-out context with a more economical "leave-a-random-chunk-out" context. In true Gibbs sampling, to resample the behavior for s_n, we use the context $\phi'_n = \sum_{m \neq n} \pi_t(s_m)$. In this training-friendly variant, we sample a_t^n for all $s_n \notin \hat{\mathcal{S}}_t$ using the shared context $\phi_t = \sum_{s \in \hat{\mathcal{S}}_t} \pi_t(s)$. The set $\hat{\mathcal{S}}_t$ is a large, random subset of $\mathcal{S}$ of size $\lfloor \gamma |\mathcal{S}| \rfloor$.

The approximation holds if $\sigma(\phi_t, s_n) \approx \sigma(\phi'_n, s_n)$ for $s_n \notin \hat{\mathcal{S}}_t$. The contexts differ by the elements in $(\mathcal{S} \setminus \{s_n\}) \setminus \hat{\mathcal{S}}_t$. This set difference has size $(|\mathcal{S}| - 1) - \lfloor \gamma |\mathcal{S}| \rfloor \approx (1 - \gamma)|\mathcal{S}| - 1$. If the fraction of held-out behaviors γ is close to 1, and the total number of contexts $|\mathcal{S}|$ is large, then ϕ_t is a very good approximation of ϕ'_n. The logic is similar to the ICM approximation. In a system with rich, distributed information, the predictive distribution is robust to small perturbations in its context. Removing a small, random fraction of context behaviors is unlikely to drastically change the inference for another behavior.

Therefore, each sampling step in this algorithm is an approximation of a true Gibbs sampling step. While not exact, the process introduces a similar dynamic that pushes the d-policy towards regions of higher coherence, and its stationary distribution should be close to the target X^β, especially for $\gamma \to 1$.

Notably, this method can be seen as an extension of the classical (and surprising) training method of **pseudo-labeling** (Lee et al., 2013).

B.3 Separating Consistent Personas from a Mixture

We test whether Gibbs sampling induces mode "snap" (convergence to a single coherent persona) instead of mode interpolation, mixture, or divergence. This is a lightly cherry-picked setup where the phenomenon is most salient; we tested two other setups whose results are directionally aligned but less consistent.

Setup. Note that this experiment measures stylistic convergence, not answer correctness, so benchmark contamination and saturation (e.g., in GSM8K-style questions) do not affect the conclusions. The experiment consists of 16 questions, including 12 scientific (e.g., "How does photosynthesis work?") and 4 emotional (e.g., "I just lost my job. . . "). We run 2048 rounds of Gibbs sampling ($2048 \times 16 = 32768$ inferences) with Claude-3.5-Haiku. Figure 7(a) displays embeddings for all generated texts over the rounds; circles for scientific questions, triangles for emotional questions; red for later rounds, blue for earlier rounds. Panels (b)–(d) show cosine similarity trends within scientific questions, within emotional questions, and between the two types.

Results. The observations are as follows: (i) scientific answers converge towards emotional answers while the latter stay put; (ii) within-type similarity first rises and then stabilizes; (iii) cross-type similarity continually rises. The implication is that the scientific mode is "snapping to" the emotional mode. The model does indeed answer scientific questions in emotional style after convergence, producing outputs like "A sacred invitation to explore the remarkable, infinite creativity of biochemical becoming" when asked the photosynthesis question. This demonstrates coherence optimization's tendency to converge on consistent personas rather than maintaining or amplifying diversity.

C Feedback-Free Self-Improvement Methods Reducible to Coherence Optimization

This section provides detailed arguments and proofs for the results showing that several state-of-the-art feedback-free self-improvement methods are reducible to coherence optimization.

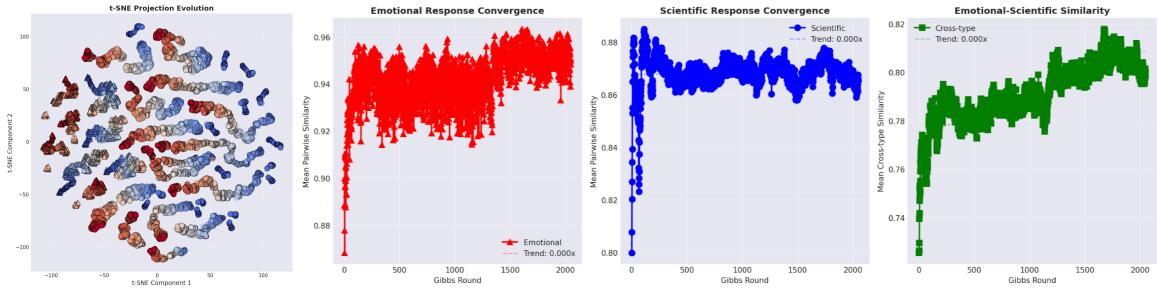

Figure 7: (a) Embeddings of scientific (circles) and emotional (triangles) answers over 2048 rounds of Gibbs sampling. Later rounds are colored red, earlier rounds blue. The model shifts to answering scientific questions in an emotional style. (b)(c)(d) Within-type and cross-type cosine similarity over 2048 rounds.

C.1 Debate

We provide the detailed argument for Proposition 4.3. Consider a learning system with two contexts, $\mathcal{S} = \{s_{\text{pro}}, s_{\text{con}}\}$. s_{pro} is the set of possible answers to "Why is P correct?" and s_{con} the set of answers to "Why is P incorrect?", for a proposition P.

If we generate arguments independently via greedy decoding, the resulting d-policy $\pi_{\text{greedy}} = (a_{\text{pro}}, a_{\text{con}})$ is likely to be incoherent. The proponent argument a_{pro} might make a claim that a_{con} trivially refutes, because a_{pro} did not anticipate this rebuttal. Generating a_{con} conditional on a_{pro} addresses only one direction of this problem.

A more robust conclusion arises from finding the d-policy $\pi^* = (a^*_{\text{pro}}, a^*_{\text{con}})$ that maximizes the *joint* coherence $\chi(\pi)$. At zero temperature ($\beta = +\infty$), this is the d-policy found by infinite-beam beam search. Such a d-policy represents a "reflective equilibrium" between the two positions, where a^*_{pro} anticipates and counters a^*_{con}, and vice versa.

This optimal d-policy can be found (or, for $\beta < \infty$, sampled from) using an iterative process that is formally identical to debate:

Algorithm 3 Iterative Debate (Gibbs Sampling for $|\mathcal{S}| = 2$)

Input: Learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S} = \{s_{\text{pro}}, s_{\text{con}}\}, \sigma)$; number of turns N; temperature reciprocal β.
Output: List of d-policies $\pi_1, \ldots, \pi_N$.
procedure DEBATE
  Independently sample $a^{(0)}_{\text{pro}} \sim \sigma^\beta(0, s_{\text{pro}})$ ▷ Proponent makes first arg
  for $t = 0, 1, \ldots, N-1$ **do**
    Independently sample $a^{(t+1)}_{\text{con}} \sim \sigma^\beta(a^{(t)}_{\text{pro}}, s_{\text{con}})$. ▷ Opponent responds to proponent's last arg
    Independently sample $a^{(t+1)}_{\text{pro}} \sim \sigma^\beta(a^{(t)}_{\text{con}}, s_{\text{pro}})$. ▷ Proponent responds to opponent's last arg
    Set $\pi_{t+1} = (a^{(t+1)}_{\text{pro}}, a^{(t+1)}_{\text{con}})$.
  end for
end procedure

This iterative refinement, where each side updates its argument based on the other's, is the essence of both debate and Algorithm 1. The convergence result (Proposition 4.3) follows directly from Theorem 4.2, noting that Algorithm 3 is a synchronous variant where each party responds to the other's latest message rather than the second-to-latest; the proof of Theorem 4.2 is agnostic to this difference.

This debate setup, which optimizes coherence over a pair of conflicting contexts, illustrates a more general principle. Ideally, we would optimize coherence over all contexts $\mathcal{S}$ simultaneously (all contexts in the world, not merely two), especially with a strong reasoning model as the prior policy. The goal would be to find a

maximally consistent system of beliefs and behaviors, a form of reflective equilibrium. This is exactly the objective of Gibbs sampling.

C.2 Simple Bootstrap Near $\beta = 1$

Algorithm 4 Simple Bootstrap

1: **Input:** Learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$; sequence of contexts $\{s_n\}_{n=1}^N$; temperature reciprocal β.
2: **Output:** Mapping $f : s_n \mapsto a_n \in s_n$.
3: **procedure** SimpleBootstrap
4: **for** $n = 1, \ldots, N$ **do**
5: Sample $a_n \sim \sigma^\beta \left(\sum_{m<n} a_m, s_n\right)$.
6: **end for**
7: **end procedure**

We provide the proof of Proposition 4.4. The proof relies on the following technical assumption.

Assumption C.1 (Context Independence on Random Subsets)**.** A learning system exhibits quantitative context independence if for every $s \in \mathcal{S}$, there exists a limiting distribution ρ_s and constants $C, K > 0, \alpha > 1$ such that for a randomly chosen sequence $(s_1, \ldots, s_{n-1})$ where $n \geq K$:

$$\mathbb{E}_{s_1,\ldots,s_{n-1}} \left[D_{\mathrm{TV}} \left(\sigma \left(\sum_{m=1}^{n-1} a_m, s_n \right), \rho_{s_n} \right) \right] \leq \frac{C}{(n-1)^\alpha}$$

for any path $(a_1, \ldots, a_{n-1})$ and any $s_n \notin \{s_1, \ldots, s_{n-1}\}$.

Proof of Proposition 4.4. We first define the two distributions to be compared. The probability of generating a d-policy $\pi = (a_1, \ldots, a_N)$ via the sequential Simple Bootstrap process is:

$$P_{\mathrm{SB}}(\pi; \beta) = \prod_{n=1}^{N} \sigma^\beta \left(\sum_{m<n} a_m, s_n \right)(a_n) = \frac{\prod_{n=1}^{N} \sigma\left(\sum_{m<n} a_m, s_n\right)(a_n)^\beta}{\prod_{n=1}^{N} Z_n\left(\sum_{m<n} a_m\right)}$$

where $Z_n(\phi) = \sum_{a' \in s_n} \sigma(\phi, s_n)(a')^\beta$ is the local normalization constant at step n.

The target softmax-over-coherence distribution is $\mathrm{X}^\beta(\pi) = \frac{1}{Z} 2^{\beta\chi(\pi)}$. Using the definition $\chi(\pi) = \sum_{n=1}^{N} \log_2 \sigma\left(\sum_{m<n} a_m, s_n\right)(a_n)$, this becomes:

$$\mathrm{X}^\beta(\pi) = \frac{1}{Z} \prod_{n=1}^{N} \sigma \left(\sum_{m<n} a_m, s_n \right)(a_n)^\beta$$

By comparing these expressions, we find that $P_{\mathrm{SB}}(\pi; \beta)$ is a reweighting of $\mathrm{X}^\beta(\pi)$:

$$P_{\mathrm{SB}}(\pi; \beta) = \mathrm{X}^\beta(\pi) \cdot w(\pi), \quad \text{where} \quad w(\pi) = \frac{Z}{V(\pi)} \quad \text{and} \quad V(\pi) = \prod_{n=1}^{N} Z_n \left(\sum_{m<n} a_m \right)$$

Since $\sum_\pi P_{\mathrm{SB}}(\pi) = 1$, taking the expectation over X^β shows that $\mathbb{E}_{\mathrm{X}^\beta}[w(\pi)] = 1$. The total variation distance is given by:

$$D_{\mathrm{TV}}\left(P_{\mathrm{SB}}, \mathrm{X}^\beta\right) = \frac{1}{2} \mathbb{E}_{\mathrm{X}^\beta}[|w(\pi) - 1|]$$

By Jensen's inequality, this is bounded by the standard deviation of the reweighting factor:

$$D_{\mathrm{TV}}^2 \leq \frac{1}{4} \mathbb{E}_{\mathrm{X}^\beta}\left[(w(\pi) - 1)^2\right] = \frac{1}{4} \mathrm{Var}_{\mathrm{X}^\beta}(w(\pi))$$

For a random variable W concentrated around its mean, its relative variance is well-approximated by the variance of its logarithm: $\mathrm{Var}(W)/\mathbb{E}[W]^2 \approx \mathrm{Var}(\ln W)$. Applying this, we find $\mathrm{Var}(w(\pi)) \approx \mathrm{Var}(\ln V(\pi))$. Our goal is to bound the variance of $\ln V(\pi)$.

Case 1: $\beta = 1$**.** In this case, $Z_n(\phi) = \sum_{a' \in s_n} \sigma(\phi, s_n)(a') = 1$ for all n and ϕ. Thus, $V(\pi) = 1$ for all π. The relation becomes $P_{\mathrm{SB}}(\pi; 1) = \mathrm{X}^1(\pi) \cdot Z$. Since both distributions must sum to 1, we must have $Z = 1$. Therefore, for $\beta = 1$, the distributions are identical, i.e., $P_{\mathrm{SB}}(\pi; 1) = \mathrm{X}^1(\pi)$.

Case 2: β **near 1.** Let $\beta = 1 + \epsilon$ for small ϵ. A first-order expansion of $\ln Z_n(\phi)$ yields:

$$\begin{aligned}
\ln Z_n(\phi) &= \ln \sum_{a'} \sigma(\phi, s_n)(a')^{1+\epsilon} \\
&= \ln \sum_{a'} \sigma(\phi, s_n)(a') \left(1 + \epsilon \ln \sigma(\phi, s_n)(a') + O\left(\epsilon^2\right)\right) \\
&= \ln \left(1 + \epsilon \sum_{a'} \sigma(\phi, s_n)(a') \ln \sigma(\phi, s_n)(a') + O\left(\epsilon^2\right)\right) \\
&= -\epsilon H(\sigma(\phi, s_n)) + O\left(\epsilon^2\right)
\end{aligned}$$

where $H(\cdot)$ is the Shannon entropy. The logarithm of the reweighting factor is therefore $\ln V(\pi) \approx -\epsilon S_H(\pi)$, where $S_H(\pi) = \sum_{n=1}^{N} H\left(\sigma\left(\sum_{m<n} a_m, s_n\right)\right)$. The variance is $\mathrm{Var}(\ln V(\pi)) \approx \epsilon^2 \mathrm{Var}(S_H(\pi))$. We now show that $\mathrm{Var}(S_H(\pi))$ is bounded by a constant independent of N.

We split the sum into a "burn-in" period ($n \leq K$) and a "stable" period ($n > K$).

In the **Stable Period** ($n > K$), the variance of this part of the sum has two components. First, we analyze the sum of the limit entropies, $\sum_{n=K+1}^{N} H(\rho_{s_n})$. Since the sequence is a random permutation, this is a sum of $N - K$ values drawn without replacement from the finite population $\{H(\rho_s)\}_{s \in \mathcal{S}}$. Its variance is $(N-K)\,\sigma_H^2 \frac{N-(N-K)}{N-1} = \frac{K(N-K)}{N-1}\sigma_H^2$, where σ_H^2 is the population variance of the limit entropies. This variance is $O(K)$ and does not grow with N. Second, we analyze the sum of the deviations, $\sum_{n=K+1}^{N} \delta_n$, where $\delta_n = H(\sigma_n) - H(\rho_{s_n})$. The variance of this sum is bounded by the sum of the individual variances (assuming weak correlation). The variance $\mathrm{Var}(\delta_n)$ is bounded by $\mathbb{E}\left[\delta_n^2\right]$, which by our assumption decays at least as fast as $O(n^{-\alpha})$. Since $\alpha > 1$, the sum $\sum_{n=K+1}^{N} \mathrm{Var}(\delta_n)$ converges to a constant as $N \to \infty$. The rate of convergence is determined by the tail of the series, which is $\sum_{n=N+1}^{\infty} n^{-\alpha} = O\left(N^{1-\alpha}\right)$. The total variance of the stable period sum is thus bounded by a constant independent of N.

In the **Burn-in Period** ($n \leq K$), the first K terms, $\sum_{n=1}^{K} H\left(\sigma\left(\sum_{m<n} a_m, s_n\right)\right)$, depend on the specific initial path and the specific initial K contexts. As entropy is a bounded function, the variance of this finite sum is bounded by a constant Δ_K^2 that depends on K but not on N.

The total variance of $S_H(\pi)$ is therefore bounded by a constant Δ^2 that is asymptotically independent of N. Combining our results:

$$D_{\mathrm{TV}} \leq \frac{1}{2}\sqrt{\mathrm{Var}(w(\pi))} \approx \frac{1}{2}\sqrt{\mathrm{Var}(\ln V(\pi))} \approx \frac{1}{2}\sqrt{\epsilon^2 \mathrm{Var}(S_H(\pi))} \leq \frac{1}{2}|\epsilon|\,\Delta$$

Since $\epsilon = \beta - 1$ and Δ is a constant for a given system, we have shown that $D_{\mathrm{TV}}\left(P_{\mathrm{SB}}, \mathrm{X}^{\beta}\right) = O(|\beta - 1|)$. □

Note that the $O(|\beta - 1|)$ asymptotics limit the usefulness of the method. The most effective use of Gibbs sampling is when $\beta \gg 1$ and the converged-upon distribution is concentrated on the highest-coherence d-policies, which would not be well-approximated by simple bootstrap.

C.3 Internal Coherence Maximization

We provide further analysis of the relationship between coherence and mutual predictability (Equations 2 and 3).

Algorithm 5 Internal Coherence Maximization (Wen et al., 2025)

1: **Input:** Learning system $(\mathcal{M}, \mathcal{A}, \mathcal{S}, \sigma)$.
2: **Output:** d-policy π^*.
3: **procedure** ICM
4: For d-policy π, define its *mutual predictability* $f_{\mathrm{MP}}(\pi) \coloneqq \sum_{n=1}^{|\mathcal{S}|} \log_2 \sigma\left(\sum_{m \neq n} \pi(s_m), s_n\right)(\pi(s_n))$.
5: Using discrete optimization methods, find $\pi^* = \arg\max_\pi f_{\mathrm{MP}}(\pi)$.
6: **end procedure**

The two objectives are approximately aligned under conditions where the learning system exhibits a form of informational saturation. The difference between the two objectives is:

$$f_{\mathrm{MP}}(\pi) - \chi(\pi) = \sum_{n=1}^{|\mathcal{S}|} \left[\log_2 \sigma\left(\sum_{m<n} \pi(s_m) + \sum_{m>n} \pi(s_m), s_n \right) - \log_2 \sigma\left(\sum_{m<n} \pi(s_m), s_n \right) \right]$$

This term measures the total change in log-probability of the behaviors $\pi(s_n)$ when informed by "future" behaviors ($\sum_{m>n} \pi(s_m)$) in addition to "past" ones.

In systems with many weakly-dependent components (large $|\mathcal{S}|$), the context for predicting any single component becomes *informationally saturated* quickly. We say that a learning system exhibits informational saturation when, for large enough context $\sum_{m<n} \pi(s_m)$, the predictive distribution $\sigma(\sum_{m<n} \pi(s_m), s_n)$ changes negligibly upon adding further context $\sum_{m>n} \pi(s_m)$. When this holds, the difference term above is small, and the d-policy π that maximizes one objective will be close to the one that maximizes the other.

This approximation is analogous to how pseudo-likelihood (based on local conditional probabilities, similar to f_{MP}) is used as a computationally cheaper proxy for the true joint likelihood (similar to χ) in fields like statistical physics and graphical models.

D Coherence Optimization as Policy-Wide Description Length Regularization

This section provides detailed proofs for the theoretical results in §5. We show that maximizing coherence imposes a policy-level regularization that improves out-of-sample generalization. As discussed in Appendix A, coherence can be further customized to regularize for high legibility, high factuality, or high alignment by appropriate choice of prior policy.

D.1 Negated Coherence as Description Length

The concept of coherence has a natural and powerful interpretation, specifically the principle of Minimum Description Length (MDL). The MDL principle states that the best model for a set of data is the one that permits the shortest description of the data.

Consider a d-policy π as a dataset, where each $\pi(s_n)$ is a data point. The coherence $\chi(\pi)$ is defined as a sum of sequential log-probabilities:

$$\chi(\pi) = \log_2 \sigma(0, s_1)(\pi(s_1)) + \log_2 \sigma(\pi(s_1), s_2)(\pi(s_2)) + \dots$$

This is precisely the log-probability of observing the sequence of behaviors $(\pi(s_1), \pi(s_2), \dots)$ under the predictive model defined by the learning system. By the chain rule, this is equal to the log of the joint probability, $\log_2 P(\pi(s_1), \dots, \pi(s_{|\mathcal{S}|}))$.

According to Shannon's source coding theorem, the minimal average number of bits required to encode a message from a source is equal to its entropy. The optimal codelength for a specific message x with probability $P(x)$ is $-\log_2 P(x)$ bits. Therefore, the negated coherence is the ideal codelength for the entire d-policy π:

$$\mathrm{CodeLength}(\pi) = -\log_2 P(\pi) = -\chi(\pi)$$

Maximizing coherence $\chi(\pi)$ is thus equivalent to minimizing the description length of the d-policy π. A d-policy is "coherent" if its constituent behaviors are highly predictable from one another, allowing for a very compact description. An "incoherent" policy consists of surprising or contradictory behaviors, requiring a longer description.

Coherence optimization, therefore, acts as a form of regularization that favors d-policies embodying strong, compressible patterns, and forces the model to discover and adhere to underlying principles rather than memorizing a set of unrelated behaviors.

D.2 Uniform Convergence Theorem for Coherence

Definition D.1 (Agreement and Accuracy)**.** For any two d-policies $\pi_1, \pi_2 \in \prod_{s\in\mathcal{S}} s$ and a subset of contexts $\hat{\mathcal{S}} \subseteq \mathcal{S}$, we define the *agreement* between π_1 and π_2 on $\hat{\mathcal{S}}$ as

$$\alpha_{\hat{\mathcal{S}}}(\pi_1;\pi_2) = \alpha_{\hat{\mathcal{S}}}(\pi_2;\pi_1) \coloneqq \frac{1}{|\hat{\mathcal{S}}|}\sum_{s\in\mathcal{S}} \mathbf{1}_{\pi_1(s)=\pi_2(s)}.$$

In particular, we denote $\alpha(\pi_1;\pi_2) \coloneqq \alpha_{\mathcal{S}}(\pi_1;\pi_2)$. Given any *ground-truth d-policy* π^*, we define d-policy π's *accuracy* on $\hat{\mathcal{S}}$ as $\alpha_{\hat{\mathcal{S}}}(\pi;\pi^*)$.

Theorem D.2 (Uniform Convergence)**.** *Let π^* be any given ground-truth d-policy, and the* training contexts $\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_N$ *be uniformly and independently sampled contexts from $\mathcal{S}$, for some $N > 0$. Let π be an arbitrary d-policy with $\alpha_{\text{train}}(\pi;\pi^*) \coloneqq \alpha_{\{\hat{s}_i\}_{i=1}^N}(\pi;\pi^*)$, then*

$$|\alpha(\pi;\pi^*) - \alpha_{\text{train}}(\pi;\pi^*)| \leq \sqrt{\frac{-2\chi(\pi) + \log_2 e - \log_2 \delta^{-1}}{2N}}$$

holds uniformly for all π with probability $1-\delta$, for any given $\delta \in (0,1)$.

Proof Sketch. The proof follows the structure of PAC-Bayesian analysis. We define a prior distribution over the space of d-policies $P(\pi) \propto 2^{\chi(\pi)}$. A standard result in learning theory (a variant of the PAC-Bayes theorem) bounds the deviation between the true risk and the empirical risk. For any specific d-policy π, we can consider a "posterior" distribution that places all its mass on π. The KL divergence between this posterior and the prior is then $-\log_2 P(\pi) \propto -\chi(\pi)$. Applying the relevant concentration inequality yields a bound of the desired form, where the complexity term penalizing generalization is the negative log-prior, i.e., the negated coherence. □

Corollary D.3 (Coherence Regularization)**.** *Under the conditions of the previous theorem, for any given set $\mathcal{R} \subseteq \prod_{s\in\mathcal{S}} s$ of d-policies and*

$$\hat{\pi} \coloneqq \arg\max_{\pi\in\mathcal{R}} \left\{ \alpha_{\text{train}}(\pi;\pi^*) - \sqrt{\frac{-2\chi(\pi) + \log_2 e - \log_2 \delta^{-1}}{2N}} \right\} \tag{9}$$

$$\bar{\pi} \coloneqq \arg\max_{\pi\in\mathcal{R}} \alpha(\pi;\pi^*), \tag{10}$$

we have

$$\alpha(\hat{\pi};\pi^*) \geq \alpha(\bar{\pi};\pi^*) - 2\sqrt{\frac{2\max(-\chi(\hat{\pi}), -\chi(\bar{\pi})) + \log_2 e - \log_2 \delta^{-1}}{2N}} \tag{11}$$

with probability $1-\delta$.

In other words, by jointly maximizing training accuracy and coherence, we can make sure the resulting policy $\hat{\pi}$ is not much worse than the optimal policy $\bar{\pi}$ when tested out-of-sample.

Remark D.4 (Coherence Regularization Matters). One may be tempted to say "But LLMs are surely already trained on sufficiently many pretraining samples? Additional regularization matters little." This is not the case.

The effective dimensionality of the pretraining corpus is too small compared to the size of model parameters. The trillions of pretraining tokens are limited in effective dimension, because (i) natural language has lots of redundancies, and, more importantly, (ii) most human speeches on the Internet can be classified into a small number of discrete "personas" that explain a large portion of the variance in the opinions and styles of those speeches.

Empirically, we see that language models, even reasoning models, are much worse at creatively synthesizing novel ideas (e.g. hypothesizing and proving new theorems) than retrieving known ones from memory, while many humans can learn creative synthesis by consuming orders-of-magnitude less data. This is a sign that models overfit to statements of human knowledge by memorizing all of them, while being less successful at discovering generalizeable reasoning strategies that led to those knowledge in the first place.

Also empirically, we see that language models frequently express mutually contradicting information or opinions in different contexts, likely due to over- and under-representation of different information sources/viewpoints in different parts of the pretraining corpus. Let us zoom out and treat a (context, response persona) pair as a training sample, where each such sample encompasses hundreds of thousands of tokens, rather than (input token sequence, output next token) as a "sample". Under such a view, the number of training samples is drastically reduced; we would like the trained policy to generalize out-of-sample (i.e., to integrate information/opinions of all personas in all contexts) instead of overfitting to training samples (i.e., to always respond in the best-represented persona of the current context); and the fact that trained language models currently do the latter suggests that additional regularization is needed.

Coherence regularization is well-placed to address both examples of overfitting above, as (i) memorization by rote and (ii) emulating best-represented persona in every context both incur a multiplicative factor on the description length of the policy, and are thus strongly punished by coherence regularization. This sets coherence regularization apart from methods such as entropy regularization, which penalizes description length of policy behavior *on each input separately*. In contrast, coherence regularization controls the ***joint*** *description length of all behaviors across all contexts.*

D.3 Good and Bad Priors for Coherence Regularization

Let us look more closely at Equation 11. We can obviously replace the regularizer $\chi(\cdot)$ with $\chi_\phi(\cdot)$ for any arbitrary prior policy ϕ, and the negative term on RHS will change accordingly. Since we want that term to be as close to 0 as possible, this gives us a measure of goodness for a prior policy.

Definition D.5 (Optimality Gap)**.** Given any ground-truth d-policy π^*, for any policy $\phi \in \mathcal{M}$, we define the *optimality gap* of ϕ as the prior policy to be

$$G(\phi; \pi^*) \coloneqq -2\chi_\phi(\pi^*) + \log_2 e > 0$$

Proposition D.6 (Optimality Gap Lower-Bounds Accuracy)**.** *Under the conditions of the uniform convergence theorem, for any policy $\phi \in \mathcal{M}$, let*

$$\hat{\pi} \coloneqq \operatorname*{arg\,max}_{\pi \in \prod_{s \in \mathcal{S}} s} \left\{ \alpha_{\text{train}}(\pi; \pi^*) - \sqrt{\frac{-2\chi_\phi(\pi) + \log_2 e - \log_2 \delta^{-1}}{2N}} \right\}, \tag{12}$$

then, with probability $1 - \delta$,

$$\alpha(\hat{\pi}; \pi^*) \geq 1 - \sqrt{\frac{2G(\phi; \pi^*) - 2\log_2 \delta^{-1}}{N}}.$$

Proof. By Equation 12, we have

$$\begin{aligned}\alpha_{\text{train}}(\hat{\pi};\pi^*) - \sqrt{\frac{-2\chi_\phi(\hat{\pi}) + \log_2 e - \log_2 \delta^{-1}}{2N}} &\geq \alpha_{\text{train}}(\pi^*;\pi^*) - \sqrt{\frac{-2\chi_\phi(\pi^*) + \log_2 e - \log_2 \delta^{-1}}{2N}} \\ &= 1 - \sqrt{\frac{-2\chi_\phi(\pi^*) + \log_2 e - \log_2 \delta^{-1}}{2N}}\end{aligned}$$

Thus,

$$\begin{aligned}\sqrt{\frac{-2\chi_\phi(\hat{\pi}) + \log_2 e - \log_2 \delta^{-1}}{2N}} &\leq \sqrt{\frac{-2\chi_\phi(\pi^*) + \log_2 e - \log_2 \delta^{-1}}{2N}} \\ -\chi_\phi(\hat{\pi}) &\leq -\chi_\phi(\pi^*)\end{aligned}$$

Applying Corollary D.3 with $\mathcal{R} = \prod_{s\in\mathcal{S}} s$ completes the proof. □

The propositions tells us that, the goodness of a prior policy is decided by the coherence of the optimal d-policy under such a prior. It's worth noting that ***any* policy can be a prior policy**, including e.g. a **(trainable) human preference model**. The job of coherence optimization is to turn any such prior policy (which, at inference time, can only evaluate the likelihood of *individual outputs/behaviors*) into a policy-level regularizer (which evaluates & regularizes the *joint* likelihood of the policy's behavior across contexts).

D.4 Proofs for Asymptotic Optimality

This section provides the proof of the Regularization Optimality Theorem (Theorem 5.5), the Change-of-Prior Lemma (Lemma 5.6), and supporting material for §5.3.

Optimality of Description-Length Regularization. Under description-length regularization, with probability $1-\delta$,

$$\begin{aligned}\alpha(\pi_{\text{SRM}}) &\geq \max_{\pi\in\mathcal{A}^{\mathcal{S}}} \alpha(\pi) - 2\,\text{reg}(\pi) \qquad \text{(uniform convergence)} \\ &\geq \mathrm{E}_{\pi\sim\mathcal{Q}}[\alpha(\pi) - 2\,\text{reg}(\pi)] \\ &\geq \mathrm{E}_{\pi\sim\mathcal{Q}}[\alpha(\pi)] - \sqrt{\frac{2}{N}}\ \mathrm{E}_{\pi\sim\mathcal{Q}}\left[\sqrt{\log_2 e/\delta - \log_2 \mathcal{P}(\pi)}\right].\end{aligned}$$

Using the Taylor expansion, we get

$$\begin{aligned}&\mathrm{E}_{\pi\sim\mathcal{Q}}\left[\sqrt{\log_2 e/\delta - \log_2 \mathcal{P}(\pi)}\right] \\ &= \mathrm{E}_{\pi\sim\mathcal{Q}}\left[\sqrt{\log_2 e/\delta} - \frac{(1+o(1))\log_2 \mathcal{P}(\pi)}{\sqrt{\log_2 e/\delta}}\right] \\ &= \sqrt{\log_2 e/\delta} - \frac{(1+o(1))}{\sqrt{\log_2 e/\delta}}\ \mathrm{E}_{\pi\sim\mathcal{Q}}\left[\log_2 \mathcal{P}(\pi)\right].\end{aligned}$$

Substituting back:

$$\alpha(\pi_{\text{SRM}}) \geq \mathrm{E}_{\pi\sim\mathcal{Q}}[\alpha(\pi)] - \sqrt{\frac{2\log_2 e/\delta}{N}} + \sqrt{\frac{2+o(1)}{N\log_2 e/\delta}}\ \mathrm{E}_{\pi\sim\mathcal{Q}}\left[\log_2 \mathcal{P}(\pi)\right].$$

Now, $\mathrm{E}_{\pi\sim\mathcal{Q}}[\log_2 \mathcal{P}(\pi)] = -\mathrm{H}[\mathcal{Q}] - \mathrm{KL}[\mathcal{Q}\|\mathcal{P}]$ by definition of cross-entropy. Simplifying:

$$\alpha(\pi_{\text{SRM}}) \geq \mathrm{E}_{\pi\sim\mathcal{Q}}[\alpha(\pi)] - \sqrt{\frac{2\log_2 e/\delta}{N}} + \sqrt{\frac{2+o(1)}{N\log_2 e/\delta}}\ \left(\mathrm{H}\left[\mathcal{Q}\right] - \mathrm{KL}\left[\mathcal{Q}\ ||\ \mathcal{P}\right]\right).$$

□

Proof of Change-of-Prior Lemma. This is a direct consequence of the definitions. We expand the terms. The left-hand side (LHS) is:

$$\text{LHS} = \sum_{l=1}^{L} \log_2 \sigma\left(\phi + \sum_{k=1}^{l-1} a_k^t, s_l^t\right)(a_l^t) + \sum_{n=1}^{N} \log_2 \sigma\left(r + \sum_{m=1}^{n-1} a_m^\phi, s_n^\phi\right)(a_n^\phi)$$

Substitute $\phi = r + \sum_{m=1}^{N} a_m^\phi$ into the first term:

$$\text{LHS} = \sum_{l=1}^{L} \log_2 \sigma\left(r + \sum_{m=1}^{N} a_m^\phi + \sum_{k=1}^{l-1} a_k^t, s_l^t\right)(a_l^t) + \sum_{n=1}^{N} \log_2 \sigma\left(r + \sum_{m=1}^{n-1} a_m^\phi, s_n^\phi\right)(a_n^\phi)$$

Now we expand the right-hand side (RHS), which is the coherence of the concatenated sequence of behaviors relative to the prior r. Denote the concatenated sequence as $\{a_i\}_{i=1}^{N+L}$, where $a_i = a_i^\phi$ for $i \leq N$ and $a_i = a_{i-N}^t$ for $i > N$.

$$\begin{aligned}\text{RHS} = \hat{\chi}_r[a_1, \cdots, a_{N+L}] &= \sum_{i=1}^{N+L} \log_2 \sigma\left(r + \sum_{j=1}^{i-1} a_j, s_i\right)(a_i) \\ &= \sum_{i=1}^{N} \log_2 \sigma\left(r + \sum_{j=1}^{i-1} a_j^\phi, s_i^\phi\right)(a_i^\phi) + \sum_{i=N+1}^{N+L} \log_2 \sigma\left(r + \sum_{j=1}^{i-1} a_j, s_i\right)(a_i)\end{aligned}$$

The first term of the RHS is exactly the second term of the LHS. Now let's examine the second term of the RHS. For $i = N + l$ (where $l \in [1, L]$), the context sum is $\sum_{j=1}^{i-1} a_j = \sum_{j=1}^{N} a_j^\phi + \sum_{j=N+1}^{N+l-1} a_j = \sum_{j=1}^{N} a_j^\phi + \sum_{k=1}^{l-1} a_k^t$. So the second term of the RHS is:

$$\sum_{l=1}^{L} \log_2 \sigma\left(r + \sum_{j=1}^{N} a_j^\phi + \sum_{k=1}^{l-1} a_k^t, s_l^t\right)(a_l^t)$$

This is exactly the first term of the LHS. Therefore, LHS = RHS. □

Example D.7 (Conditional Probabilities in Bayesian Learning System)**.** In a Bayesian learning system, Theorem 5.8 becomes a simple identity of log-probabilities. The base coherence $\chi(\pi)$ is the log joint probability of all behaviors in the d-policy π: $\chi(\pi) = \log_2 P(\pi(\mathcal{S})) = \log_2 P(\pi(\mathcal{S}_a), \pi(\mathcal{S}_b))$. The "pretrain coherence" $\hat{\chi}_0[\pi(\mathcal{S}_b)]$ is the log marginal probability of the pretraining behaviors: $\hat{\chi}_0[\pi(\mathcal{S}_b)] = \log_2 P(\pi(\mathcal{S}_b))$. The "posttrain coherence w.r.t. pretrain prior" $\chi^a(\pi(\mathcal{S}_a))$ corresponds to coherence in a quotient system where the base policy is $0_a = \sum_{s \in \mathcal{S}_b} \pi(s)$. This is equivalent to starting with a prior conditioned on the pretraining data $\pi(\mathcal{S}_b)$. Thus, this term is the log conditional probability: $\chi^a(\pi(\mathcal{S}_a)) = \log_2 P(\pi(\mathcal{S}_a)|\pi(\mathcal{S}_b))$. Plugging these into the first half of Equation 7:

$$\underbrace{\log_2 P(\pi(\mathcal{S}_a)|\pi(\mathcal{S}_b))}_{\chi^a(\pi(\mathcal{S}_a))} + \underbrace{\log_2 P(\pi(\mathcal{S}_b))}_{\hat{\chi}_0[\pi(\mathcal{S}_b)]} = \underbrace{\log_2 P(\pi(\mathcal{S}_a), \pi(\mathcal{S}_b))}_{\chi(\pi)}$$

This is simply the chain rule of probability, $\log_2 P(A|B) + \log_2 P(B) = \log_2 P(A, B)$. The other half of the equation follows from symmetry, as $P(A, B) = P(B, A) = P(B|A)P(A)$, which corresponds to the right-hand side of Equation 7.

D.5 Heuristic Argument for the Equivalence Between Regularization and Optimization

We now informally show why the two formulations in Conjecture 5.9 are likely equivalent under appropriate conditions. By Equation 7, optimizing coherence on the posttrain samples $\pi(\mathcal{S}_a)$ with respect to a pretrained policy is equivalent to jointly optimizing (i) the coherence of posttrain samples with respect to zero prior

and (ii) the accuracy of a posttrain-only policy on pretrain samples. (Since $\pi(\mathcal{S}_b)$ is fixed, we may ignore the "pretrain coherence" term.)

This looks similar to the regularized training objective from Proposition 5.4, but there are two gaps to bridge. First, $\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]$ appears linearly in the coherence objective, while it appears under a square root in the regularization term $\sqrt{[-2\chi(\pi)+\log_2 e-\log_2(1/\delta)]/2N}$. Second, $\chi^b(\pi(\mathcal{S}_b))$ is coherence on pretrain samples, not accuracy $\alpha_{\text{train}}(\pi)$.

We address both gaps with heuristic calculations.

Let's first look at the second question. We know that in a naive Bayes model, the log likelihood scales linearly with accuracy times sample size. In supervised training, this is approximately true up to a constant factor. In LLM finetuning pipelines, the cross-entropy loss typically drops by around a factor of 2 before stagnating, so the cumulative loss is approximately linear up to a factor of 2. The same is true of pretraining loss after the initial stage of rapid loss decline. We may thus assume

$$\left|c|\mathcal{S}_b|(\alpha^b(\pi)-1)-\chi^b(\pi(\mathcal{S}_b))\right| \leq \epsilon|\mathcal{S}_b|$$

for some constants $c>0, d<0$ and small ϵ. This answers the second question. After we multiply the pretrain accuracy by $c|\mathcal{S}_b|$, the two are approximately equal in size.

Now, the first question. Observe that Proposition 5.4 does not rely on the accuracy and coherence being calculated on the same context set, and not even that the two being calculated on the same number of contexts. We only require that *the set of contexts we use for specifying the trained d-policy be included in the calculation of coherence.* We can thus switch the $\chi(\pi)$ in the regularization term for $\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]$ (where $\mathcal{S}_a$ is the "set of contexts we use for specifying d-policy"), and N, on the other hand, for $|\mathcal{S}_b|$ (the number of samples used for training accuracy calculation). We must also multiply the regularization term by the same factor of $c|\mathcal{S}_b|$, so we are now comparing $\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]$ and

$$c|\mathcal{S}_b|\sqrt{\frac{-2\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]+\log_2 e-\log_2(1/\delta)}{2|\mathcal{S}_b|}} = c\sqrt{|\mathcal{S}_b|}\cdot\sqrt{-\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]+\frac{\log_2 e-\log_2(1/\delta)}{2}}$$

Now we have an oracle that, when given any $\mathcal{S}_a, \mathcal{S}_b, \pi(\mathcal{S}_b)$, can tell us the optimum for

$$\begin{aligned}
&\max_{\pi(\mathcal{S}_a)}\left(c|\mathcal{S}_b|\alpha^b(\pi)+\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]\right)\\
=&\max_{r\in\mathbb{R}}\left(\max_{\pi(\mathcal{S}_a):\ -\hat{\chi}_0[\pi(\mathcal{S}_a)]/|\mathcal{S}_a|=r} c|\mathcal{S}_b|\alpha^b(\pi)-|\mathcal{S}_a|\cdot r\right)\\
:=&\max_{r\in\mathbb{R}}(f(r)-|\mathcal{S}_a|\cdot r).
\end{aligned}$$

while we would like to solve for the regularized accuracy, i.e., roughly

$$\begin{aligned}
&\max_{\pi(\mathcal{S}_a)}\left(c|\mathcal{S}_b|\alpha^b(\pi)-c\sqrt{|\mathcal{S}_b|}\cdot\sqrt{-\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]+\frac{\log_2 e-\log_2(1/\delta)}{2}}\right)\\
=&\max_{r\in\mathbb{R}}\left(\max_{\pi(\mathcal{S}_a):\ -\hat{\chi}_0[\pi(\mathcal{S}_a)]/|\mathcal{S}_a|=r} c|\mathcal{S}_b|\alpha^b(\pi)-c\sqrt{|\mathcal{S}_b|}\cdot\sqrt{|\mathcal{S}_a|\cdot r+\frac{\log_2 e-\log_2(1/\delta)}{2}}\right)\\
\approx&\max_{r\in\mathbb{R}}\left(\max_{\pi(\mathcal{S}_a):\ -\hat{\chi}_0[\pi(\mathcal{S}_a)]/|\mathcal{S}_a|=r} c|\mathcal{S}_b|\alpha^b(\pi)-\sqrt{|\mathcal{S}_a|}\cdot c\sqrt{|\mathcal{S}_b|}\cdot\sqrt{r}\right)\\
:=&\max_{r\in\mathbb{R}}\left(f(r)-\sqrt{|\mathcal{S}_a|}\cdot g(r)\right).
\end{aligned} \tag{13}$$

Note that, since $-\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]$ is approximately asymptotically linear with respect to $|\mathcal{S}_a|$, by defining r with $-\hat{\chi}_0\left[\pi(\mathcal{S}_a)\right]/|\mathcal{S}_a|$, we ensure that the function $f(r)$ doesn't vary by orders of magnitude when $|\mathcal{S}_a|$ changes. We will now assume $f(r)$ is a fixed function independent of $|\mathcal{S}_a|$, which is true when $|\mathcal{S}_a|$ is large and the ratio r has converged for every possible d-policy.

$g(r)$ is concave. Empirically, $f(r)$ is also likely approximately concave, as it represents the tradeoff between coherence ($\approx$ accuracy) on $\mathcal{S}_a$ and accuracy on $\mathcal{S}_b$, and accuracy tradeoff on two different datasets likely exhibits diminishing returns.

Now, remember that there's one more variable we can control, but which we've so far not put to use: $|\mathcal{S}_a|$. We will now use it as a Lagrange multiplier, and denote it with λ.

Given any λ, the oracle gives us the optimal $r^* : f'(r^*) = \lambda$. Our hope is to find a value for λ such that $r^*(\lambda)$ also satisfies the optimality condition for Equation 13, i.e.,

$$f'(r^*) - \sqrt{\lambda} \cdot g'(r^*) = \lambda - \sqrt{\lambda} \cdot g'(r^*) = \lambda - \sqrt{\lambda} \cdot \frac{1}{2} c \sqrt{\frac{|\mathcal{S}_b|}{r^*}} = 0 \iff \lambda = \frac{c^2}{4r^*}|\mathcal{S}_b|$$

Now we know that such a $|\mathcal{S}_a|^*$ exists and

$$\begin{aligned}
|\mathcal{S}_a|^* &\approx \frac{c^2}{4r^*}|\mathcal{S}_b| \\
&\approx \frac{\left(-\chi^b(\pi^*(\mathcal{S}_b))/[|\mathcal{S}_b|(1-\alpha^b(\pi^*))]\right)^2}{4(-\hat{\chi}_0[\pi^*(\mathcal{S}_a)])/|\mathcal{S}_a|^*}|\mathcal{S}_b| \\
&= \frac{1}{4} \times \underbrace{\frac{\overbrace{\left(-\chi^b(\pi^*(\mathcal{S}_b))/|\mathcal{S}_b|\right)^2}^{\text{mean pretrain coherence, squared}}}{-\hat{\chi}_0[\pi^*(\mathcal{S}_a)]/|\mathcal{S}_a|^*}}_{\text{mean posttrain coherence, inversed}} \times \underbrace{\left(\frac{1}{1-\alpha^b(\pi^*)}\right)^2}_{\text{pretrain error rate, inverse-squared}} \times \underbrace{|\mathcal{S}_b|}_{\text{pretrain sample count}} .
\end{aligned}$$

E Out-of-Sample Evaluation of Gibbs Sampling on GSM8K

We report out-of-sample (test) accuracy for the training-friendly Gibbs sampling experiment from §6. The original experiment only reported in-sample (train) accuracy. Here we use the same setup with an expanded evaluation. We run 100 seeds per γ value, with 100 train questions and 200 held-out test questions from GSM8K. The model is `meta-llama/Llama-3.2-1B-Instruct`, the temperature is 0.3, and we run 40 Gibbs rounds for each $\gamma \in \{0.25, 0.45, 0.65, 0.85\}$. The test questions are never included in the Gibbs sampling context; they are only used for evaluation.

Results. Table 2 reports the initial, final, and peak accuracies for both train and test splits. Figure 8 shows the train and test accuracy trajectories over Gibbs rounds.

We could make several observations. First, test accuracy improves substantially for all γ values, rising from roughly 30% at initialization to 38–40% at the final round. The mean peak (best round of the seed-averaged trajectory) reaches 39–41%, while the oracle peak (average of per-seed maxima, where each seed selects its own best round) reaches 46–47%. The gap between these two numbers reflects the noisiness of individual runs. Both confirm that coherence optimization generalizes to held-out questions rather than merely memorizing the in-sample context.

Second, the train-test gap behaves differently across γ. At $\gamma = 0.25$ (small hold-out fraction, strongest resampling), the mean peak reaches 45.7% train vs. 39.3% test, a gap of about 6 percentage points. At $\gamma = 0.85$ (large hold-out fraction, weakest resampling), the gap essentially vanishes (40.7% train vs. 40.4% test). Higher γ values sacrifice some in-sample performance but generalize more evenly, consistent with the regularization interpretation.

Third, while in-sample accuracy varies considerably with γ (from 40.4% to 44.9% at round 40), test accuracy is remarkably stable across γ values (from 38.1% to 40.2% at round 40, and from 39.3% to 41.2% at the mean peak). This suggests that the generalization benefit of coherence optimization is robust to the choice of γ, even though the in-sample signal is stronger at lower γ.

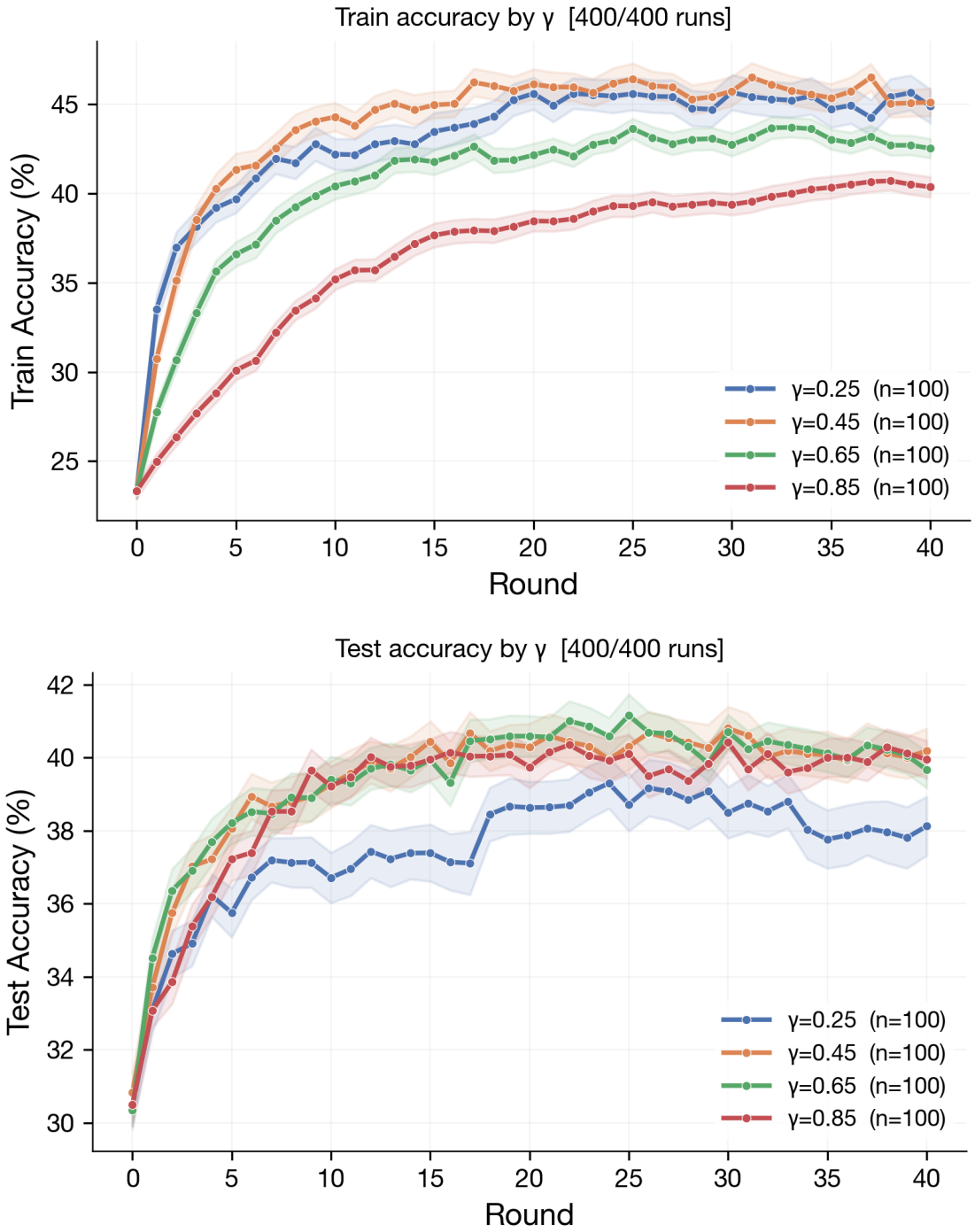

Figure 8: Train (top) and test (bottom) accuracy over 40 Gibbs rounds for $\gamma \in \{0.25, 0.45, 0.65, 0.85\}$, averaged over 100 seeds per γ (shaded region: 95% CI).

Table 2: Train and test accuracy over 40 Gibbs rounds, averaged over 100 seeds per γ ($\pm$ SE). We report two notions of peak accuracy. **Mean peak** first averages accuracy across seeds at each round, then takes the maximum of this averaged trajectory. **Oracle peak** takes the maximum accuracy across rounds for each seed individually, then averages these per-seed maxima. The oracle peak is higher because each seed selects its own best round.

γ	Init (%)	Final (%)	Mean peak (%)	Oracle peak (%)	Improv. (%)
Train (100 questions, in-sample)					
0.25	23.3 ± 0.4	44.9 ± 1.0	45.7 (r30)	56.2 ± 0.6	$+21.6 \pm 1.0$
0.45	23.3 ± 0.4	45.1 ± 0.8	46.5 (r31)	54.9 ± 0.6	$+21.8 \pm 0.9$
0.65	23.3 ± 0.4	42.5 ± 0.5	43.7 (r33)	50.6 ± 0.5	$+19.2 \pm 0.6$
0.85	23.3 ± 0.4	40.4 ± 0.6	40.7 (r38)	45.2 ± 0.5	$+17.0 \pm 0.7$
Test (200 questions, out-of-sample)					
0.25	30.5 ± 0.6	38.1 ± 0.8	39.3 (r24)	46.6 ± 0.4	$+7.7 \pm 0.9$
0.45	30.8 ± 0.6	40.2 ± 0.6	40.8 (r30)	47.0 ± 0.4	$+9.4 \pm 0.8$
0.65	30.4 ± 0.5	39.7 ± 0.5	41.2 (r25)	46.9 ± 0.4	$+9.3 \pm 0.7$
0.85	30.5 ± 0.6	40.0 ± 0.5	40.4 (r30)	46.1 ± 0.4	$+9.5 \pm 0.8$