UNCOVERING LATENT REASONING STRATEGIES IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

A language model $p_\theta(y|x)$ trained on reasoning tasks learns to solve problems via multiple distinct strategies, yet these strategies are implicit and entangled within the model's response distribution. We study the problem of decomposing the response distribution of a given pretrained language model into a structured, strategy-conditioned representation. Specifically, we learn a latent-variable factorization $p_\theta(y|x) \rightsquigarrow (r_\phi(z|x),\, g_\phi(y|x,z))$, where a router r maps each input to a distribution over latent strategies z and a generator g produces the response conditioned on that strategy. A key challenge is that the generator, initialized from the base model, already represents $p_\theta(y|x)$ without using z. Standard variational inference therefore gives the model no incentive to route information through z and can yield a severe form of posterior collapse. To address this, we propose a variational objective that measures fractional information gain relative to the base model's response loss and concentrates reconstruction pressure on tokens with high base model surprisal, encouraging z to encode strategy-relevant response variation. We introduce a benchmark of multi-strategy algorithmic tasks and show that this objective recovers latent codes aligned with distinct reference strategies while preserving the base model's response distribution. On this controlled benchmark, we show that active intervention in the learned latent space enables more efficient exploration of the model's strategy space.

1 INTRODUCTION

Language models trained on reasoning tasks can solve the same problem using multiple distinct strategies. For example, a model asked to prove a theorem may produce a proof by contradiction, induction, or construction, while a model asked to solve a programming problem may formulate it as dynamic programming, graph search, or a greedy algorithm. These strategies can recur across many problems, but are represented only implicitly. The model's response distribution $p_\theta(y|x)$ is an entangled mixture over distinct strategies, with no explicit variable for identifying or controlling which strategy is used.

We study how to uncover these strategies by decomposing the response distribution of a pretrained language model into a structured, strategy-conditioned representation. Concretely, we seek a *router-generator factorization* $(r_\phi(z|x),\, g_\phi(y|x,z))$ of the base model $p_\theta(y|x)$, where a router maps each input to a distribution over a latent strategy variable z, and a strategy-conditioned generator produces responses given the input and the sampled strategy. The goal is to find a factorization that preserves the observable response distribution, so that $\int r_\phi(z|x) g_\phi(y|x,z) dz =: p_\phi(y|x) \approx p_\theta(y|x)$, while making z informative and strategy-aligned: varying z should change which strategy the generator uses, and the same latent value should carry the same strategy meaning across related inputs. Figure 1 illustrates this target: the base model mixes strategies implicitly, a latent-variable model initialized to match it leaves z unused, and the factorization we seek separates strategies in latent space while preserving the response distribution.

This problem is distinct from standard latent variable generative modeling (Kingma and Welling, 2013; Sohn et al., 2015), where the model is trained from data and the target distribution is not already represented at initialization. Here, the pretrained generator already captures $p_\theta(y|x)$ without any latent variable, creating a pathological form of posterior collapse. A factorization that ignores z, reproducing the base model for every z, is available at initialization and is globally optimal for any objective that only fits the observable response distribution. Standard variational training is therefore subject to this failure mode (Bowman et al., 2016; He et al., 2019; Fu et al., 2019; Hoffman and Johnson, 2016), as it can preserve distributional fidelity while leaving z inert.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

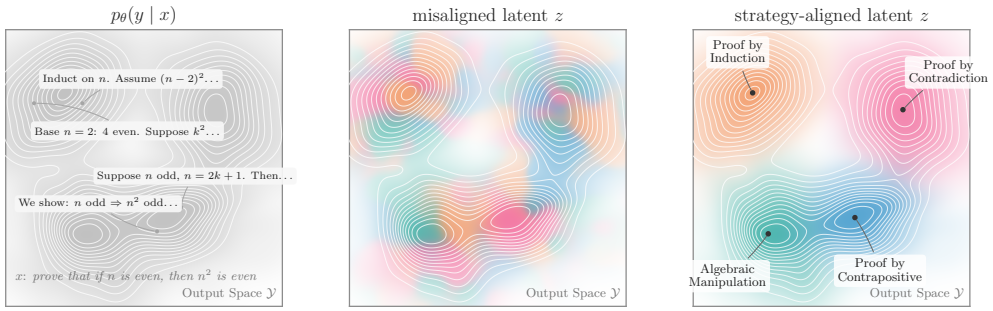


Figure 1: **Latent strategy structure in output space.** A pretrained LM's response distribution mixes strategies implicitly (left). A latent variable model initialized to match it leaves z unused (center). Training aligns z with reference strategies, disentangling them in latent space while preserving the base model's response distribution (right).

To address this, we propose a training objective that replaces marginal-fit pressure with *model-directed reconstruction pressure*. The fixed base model p_θ serves as a reference for what is already predictable from x alone. The reconstruction term is normalized by the base model's response loss and weighted at the token level by base model surprisal, concentrating pressure on branch points where multiple continuations remain plausible and strategy choices tend to be made. This encourages z to encode strategy-relevant variation that is not already determined by x.

More broadly, this problem offers a way to expose latent capabilities that language models already possess but do not reliably express under default sampling. A strategy-aligned factorization turns implicit strategy structure into an explicit object of analysis and intervention, with applications to interpretability (Koh et al., 2020), controllability (Keskar et al., 2019; X. L. Li and Liang, 2021), and exploration (Wang et al., 2023; Yao et al., 2023). In particular, strategy-level interventions could support directed exploration over high-level solution strategies during reinforcement-learning post-training, rather than passive on-policy sampling from the base model (Yue et al., 2025; Tuyls et al., 2025), helping discover under-sampled strategies and transfer them across inputs.

We summarize the contributions of this paper as follows:

1. **Problem formulation.** We formulate the problem of learning a router-generator factorization of a pretrained language model and identify the collapse problem that distinguishes it from standard latent variable modeling.
2. **Methodology.** We propose a variational objective that uses the base model to normalize reconstruction loss and to direct token-level pressure toward high-surprisal positions, encouraging the latent to encode strategy-relevant variation.
3. **Benchmark and empirical evaluation.** We introduce a benchmark of multi-strategy algorithmic tasks with ground-truth strategy annotations. We show that standard variational objectives leave z inert even though strategy information is decodable from the base model's hidden states, whereas our objective recovers latents that are strategy-aligned and consistent across inputs while preserving the base model's response distribution. Intervening in the learned latent space also improves the efficiency of strategy exploration at a fixed sampling budget.
4. **Theoretical analysis.** We initiate a theoretical analysis of strategy-structure recovery, giving sufficient conditions under which a complexity-penalized variational objective identifies a target strategy structure.

2 Problem Statement: Latent-Variable Factorization

Let $p_\theta(y|x)$ denote the response distribution of a pretrained base model, where x is the input and y is the response sequence. We consider reasoning tasks for which $p_\theta(y|x)$ mixes multiple high-level strategies for producing valid responses. The goal of this paper is to expose such variation as an explicit latent variable while preserving the observable response distribution that the base model already represents. We formalize this goal as a *latent-variable router-generator factorization* of $p_\theta(y|x)$: the router $r(z|x)$ maps each input to a distribution over latent strategy values z, and the strategy-conditioned generator $g(y|x, z)$ produces responses conditioned on both x and z.

Problem Statement: Latent-variable factorization of a pretrained model

Let $p_\theta(y|x)$ be the fixed response distribution of a pretrained base model. A router-generator factorization $p_\theta(y|x) \rightsquigarrow p_\phi = (r_\phi, g_\phi)$ introduces a latent variable $z \in \mathcal{Z}$, a router $r_\phi(z|x)$, and a strategy-conditioned generator $g_\phi(y|x, z)$, inducing

$$p_\phi(y|x) = \int_{\mathcal{Z}} r_\phi(z|x) g_\phi(y|x, z)\, d\mu(z).$$

The goal is to select a factorization that preserves the observable response distribution, $p_\phi(y|x) \approx p_\theta(y|x)$, while making z informative about y beyond x and strategy-aligned: variation in z should expose high-level, reusable strategy variation in responses rather than arbitrary predictive surface details.

The objective cannot be merely to fit the latent marginal $p_\phi(y|x)$ to the base model distribution. For expressive router-generator classes, many factorizations can induce nearly the same distribution over y after integrating out z. One extreme ignores z, with $g_\phi(y|x, z)$ reproducing the base model distribution for all z. More generally, a marginal-preserving factorization may use z for input-specific partitions, surface form, or other predictive details rather than reusable strategy structure. The problem is therefore not ordinary latent-variable modeling of $p_\theta(y|x)$, but selection among marginal-preserving factorizations for one whose latent variable is informative and strategy-aligned.

The desired latent must first be informative: z should encode information about y beyond what is already determined by x. This is nontrivial for autoregressive language models, where the base model already defines a stochastic response distribution without an explicit latent; making z informative therefore requires it to encode structured variation across responses. However, informativeness alone is insufficient: a latent can help predict y by encoding surface-level detail or input-specific partitions. We seek an informative latent that is *strategy-aligned*, where a strategy is a high-level, reusable attribute of a response trace: the broad procedure used to generate a solution, not incidental wording, formatting, or token-level variation. A strategy-aligned factorization is one in which varying z varies the generator along this strategy-level axis, while marginalizing over z recovers the base model.

There are two distinct requirements: local strategy separation and stable cross-input latent semantics. Within-input strategy separation asks whether varying z changes the strategy used for a fixed input x. This rules out an inert latent, but not an input-local code whose interpretation changes across problems. The stronger requirement is cross-input semantic consistency: a latent value or region should retain its strategy meaning across related inputs. Only with this stability does z behave like a reusable strategy variable, rather than an input-specific response index.

To measure whether a factorization succeeds, we use controlled experiments in which response traces are parseable and their reference strategy can be classified. These strategy labels are withheld during factorization training and used only to evaluate whether the learned latent achieves within-input strategy separation and cross-input semantic consistency. The next section develops a procedure for learning a router-generator factorization whose latent variable is informative and strategy-aligned.

3 Learning Strategy-Aligned Latent-Variable Factorizations

This section describes our methodology for learning a router-generator factorization of a pretrained base model whose latent variable is informative and strategy-aligned. We first describe the architecture of the router-generator model, then show why standard variational inference leads to latent collapse in this setting, and finally introduce a training objective that addresses this by using the base model to direct reconstruction pressure. Throughout, $p_\theta(y|x)$ denotes the fixed pretrained base model, ϕ the parameters of the router-generator model, and ξ the parameters of the training-time posterior.

3.1 A Transformer Architecture for Router-Generator Factorization

We implement the latent-variable router-generator factorization by *adapting* a pretrained Transformer language model into the three roles shown in Figure 2. The router $r_\phi(z|x)$ reads the input and produces a distribution over latent strategy values. Sampling from it selects a strategy for generating a response. The strategy-conditioned generator $g_\phi(y|x, z)$ generates the response autoregressively while conditioning on x and an embedding $E(z)$ of the latent strategy. The training-time posterior $q_\xi(z|x, y)$ reads the full input-response pair and infers a distribution over latent values, providing the variational training signal.

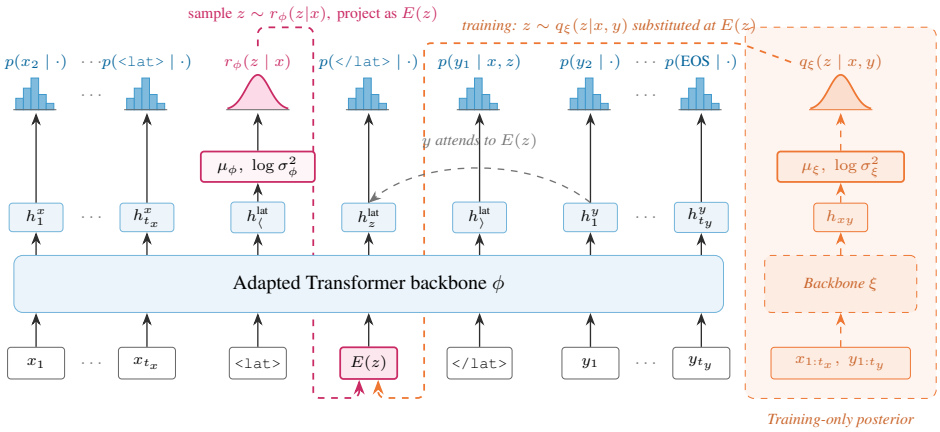


Figure 2: **Autoregressive router-generator Transformer architecture.** The adapted Transformer ϕ implements both the router $r_\phi(z|x)$, which reads the prompt and produces a latent distribution, and the strategy-conditioned generator $g_\phi(y|x,z)$, which attends to $E(z)$ while generating the response. During variational training, $q_\xi(z|x,y)$ infers z from the full input-response pair.

This parameterization requires only small architectural additions to the base model. Let $h^{(L)}_\phi(x)$ denote the final-layer representation produced by the adapted Transformer after reading the input prefix x. The router adds two learned projection heads on top of this representation, producing the mean and log-variance of a diagonal Gaussian over the continuous latent:

$$r_\phi(z|x) = \mathcal{N}\Big(W_\mu h^{(L)}_\phi(x), \operatorname{diag}\Big(\exp(W_\sigma h^{(L)}_\phi(x))\Big)\Big)\,.$$

During autoregressive generation, a latent sample z is projected to an embedding $E_\phi(z)$ and inserted as a pseudo-token at the embedding layer, so response tokens can attend to it while the generator predicts y. The training-time posterior $q_\xi(z|x,y)$ is implemented similarly to the router, parameterizing a Gaussian distribution over z from the final-layer input representation after reading (x,y).

The router and generator are a single autoregressive model with shared parameters ϕ. The posterior is a separate model with parameters ξ, and is used only during variational training. Both models are initialized from the base parameters θ and trained as light adaptations using LoRA-style low-rank updates (Hu et al., 2022), together with the small Gaussian heads and latent embedding projection.

Because $p_\theta(y|x)$ is already represented by the base model, lightweight adaptation lets the generator inherit the base model's response distribution rather than relearn it. However, because the base generator realizes the target response distribution, a factorization that ignores z is available at initialization, biasing optimization toward it when the objective only measures marginal fit. The training objective must therefore choose a factorization where z explains meaningful strategy variation.

3.2 ELBO Under-Specification and Latent Collapse

The standard negative conditional ELBO objective for this router-generator model is

$$\mathcal{J}_{\text{cELBO}}(x,y;\phi,\xi) = \mathop{\mathbb{E}}_{z\sim q_\xi(\cdot|x,y)}\left[-\log g_\phi(y|x,z)\right] + \mathrm{KL}(q_\xi(\cdot|x,y)\,\|\,r_\phi(\cdot|x))\,. \tag{1}$$

Let $p_\phi(z|x,y) = r_\phi(z|x)g_\phi(y|x,z)/p_\phi(y|x)$ denote the posterior induced by the router-generator factorization. This objective satisfies the identity

$$\mathcal{J}_{\text{cELBO}}(x,y;\phi,\xi) = -\log p_\phi(y|x) + \mathrm{KL}(q_\xi(\cdot|x,y)\,\|\,p_\phi(\cdot|x,y))\,. \tag{2}$$

Thus the standard variational objective asks the induced marginal $p_\phi(y|x) = \int_{\mathcal{Z}} r_\phi(z|x)g_\phi(y|x,z)\,d\mu(z)$ to match the observed response distribution, and asks q_ξ to match the posterior of whichever factorization produced that marginal. It does not ask whether z explains variation in y beyond x, nor whether the resulting latent is strategy-aligned.

This is an objective-level under-specification problem. If the generator class can represent the base model while ignoring z, then the factorization

$$g_\phi(y|x,z) = p_\theta(y|x) \quad \text{for all } z, \qquad q_\xi(z|x,y) = r_\phi(z|x)$$

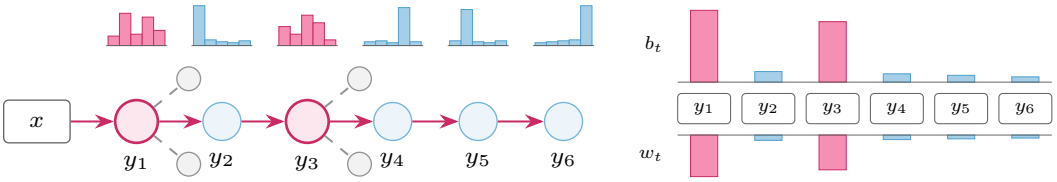


Figure 4: **Token-level p_θ surprisal weighting.** The left panel shows a response path with branch points, where several next tokens remain plausible, and determined positions, where the next token is mostly fixed by x and earlier response tokens. The right panel shows z-free base model surprisal $b_{\theta,t}(x, y)$ above each token and the corresponding token weight below it; higher-surprisal positions receive more reconstruction pressure in $a_{\theta,t}(x, y)$.

preserves the base model's response distribution and attains the objective's optimum, yet leaves the latent inert even though $p_\phi(y|x) = p_\theta(y|x)$. With pretrained generators, this solution is also close to initialization, since the generator need only learn to ignore z. The resulting latent collapse is driven by both the objective's under-specification and the proximity of the inert solution to initialization.



Figure 3: **ELBO training can preserve fidelity while leaving the latent unused.** Distributional fidelity rises, posterior-router KL collapses relative to initialization, and strategy alignment remains low.

3.3 Model-Directed Reconstruction for Informative, Strategy-Aligned Latents

The ELBO failure above shows that marginal fidelity is not enough: a factorized model can preserve $p_\theta(y|x)$ while leaving z unused. We address this selection problem with a *model-directed reconstruction term*. The base model p_θ is used to concentrate reconstruction pressure on response tokens that carry high information content under p_θ, so the latent is encouraged to explain variation in y that the base model does not already explain through x and the autoregressive prefix.

The fixed base model defines the response distribution being factorized, and training examples are drawn as

$$x \sim \mathcal{D}_X, \qquad y \sim p_\theta(\cdot|x), \qquad z \sim q_\xi(\cdot|x, y).$$

Thus y is sampled from the base model, while z is inferred by the training-time posterior for that sampled input-response pair. No strategy annotations are available during training.

The core methodological idea is to use the base model p_θ to *adaptively* rescale the reconstruction loss and direct reconstruction pressure towards tokens whose realized value carries high information content under the base model. We instantiate this idea with token-level base model surprisal. Define

$$b_{\theta,t}(x, y) = -\log p_\theta(y_t|x, y_{<t}), \qquad b_\theta(x, y) = \frac{1}{T_y}\sum_{t=1}^{T_y} b_{\theta,t}(x, y),$$

as the z-free base model surprisal. This score is computed by the latent-free base model. $b_{\theta,t}(x, y)$ is the information content of the realized token y_t given x and $y_{<t}$, and therefore gives a local estimate of where z can explain variation beyond the z-free prefix model. Its conditional expectation over $y_t \sim p_\theta(\cdot|x, y_{<t})$ is the next-token entropy, so $b_{\theta,t}$ is a path-specific sample of the base model's predictive uncertainty. We use the realized quantity rather than the entropy because the posterior conditions on the sampled response, and the latent is meant to explain the continuation that occurred.

This token-level view matters because autoregressive responses quickly become partially self-predictable. After an early commitment, many later tokens may be determined by x and $y_{<t}$, even if the commitment itself reflects a strategy choice. Our reconstruction term therefore replaces uniform token pressure with p_θ-surprisal-directed pressure, assigning more weight to branch points and high-surprisal positions where the base model leaves multiple continuations plausible. Figure 4 illustrates this weighting on a schematic response path.

The proposed objective is

$$\begin{aligned}
\mathcal{J}_\theta(x, y; \phi, \xi) &= \mathcal{R}_\theta(x, y; \phi, \xi) + \beta \, \mathrm{KL}(q_\xi(\cdot|x, y) \,\|\, r_\phi(\cdot|x)) \,, \\
\mathcal{R}_\theta(x, y; \phi, \xi) &= \frac{1}{c_\theta} \mathop{\mathbb{E}}_{z \sim q_\xi(\cdot|x,y)} \left[\sum_{t=1}^{T_y} a_{\theta,t}(x, y) \left[-\log g_\phi(y_t|x, z, y_{<t}) \right] \right] , \\
a_{\theta,t}(x, y) &= \alpha \, \frac{1}{T_y} + (1 - \alpha) \, w^{(\gamma)}_{\theta,t}(x, y), \qquad c_\theta = \mathop{\mathbb{E}}_{x \sim \mathcal{D}_X, \; y \sim p_\theta(\cdot|x)} \left[b_\theta(x, y) \right] .
\end{aligned} \tag{3}$$

A θ subscript marks quantities that depend on the base model p_θ. The global scale c_θ measures reconstruction in base model reference units, keeping it comparable to the KL term across data distributions, even when the base model's z-free loss is already low. The parameter $\alpha \in [0, 1]$ mixes uniform token pressure with surprisal-directed pressure, using token weights $w^{(\gamma)}_{\theta,t}(x, y)$ defined as

$$\widetilde{w}^{(\gamma)}_{\theta,t}(x, y) = \frac{b_{\theta,t}(x, y)^\gamma}{\sum_{s=1}^{T_y} b_{\theta,s}(x, y)^\gamma}, \qquad w^{(\gamma)}_{\theta,t}(x, y) = \frac{\widetilde{w}^{(\gamma)}_{\theta,t}(x, y)}{\kappa^{(\gamma)}_\theta(x, y)}, \tag{4}$$

where $\kappa^{(\gamma)}_\theta(x, y) = \frac{1}{b_\theta(x,y)} \sum_{s=1}^{T_y} \widetilde{w}^{(\gamma)}_{\theta,s}(x, y) \, b_{\theta,s}(x, y)$ is a normalization factor that preserves the z-free base scale of the reconstruction term, and $\gamma \geq 0$ controls the concentration of the token weights. By design, $\mathbb{E}_{x,y} \left[\mathcal{R}_\theta(x, y; \phi, \xi) \right] = 1$ when $g_\phi(y_t|x, z, y_{<t}) = p_\theta(y_t|x, y_{<t})$, so $\mathcal{R}_\theta$ is the (weighted) fraction of the base model's response loss that remains after conditioning on z. The objective thus trades off the KL penalty against the *fraction* of base model loss explained by z, giving a fractional information-gain criterion rather than one in absolute log-loss units.

4 Experiments

4.1 Benchmark and evaluation protocol

We evaluate on a controlled benchmark of multi-strategy algorithmic tasks designed to make strategy alignment measurable. Because the tasks are parseable, generated traces can be validated and matched to benchmark strategies, giving ground-truth evaluation of whether a learned latent tracks strategy-level variation. Each example consists of a problem instance x and a solution trace y. The benchmark covers six task families: list summation, sorting algorithms, grid pathfinding, linear equation solving, base conversion, and multidigit addition. Solution traces instantiate task-specific procedures for reaching the same valid answer, such as different traversal orders, algebraic manipulation orders, or algorithms. Details of the benchmark design are provided in Appendix C.

We use a two-stage experimental protocol. First, to construct a controlled factorization target with known reference strategies, we fit a language model to the benchmark distribution, producing a base model $p_\theta(y|x)$ whose response distribution mixes multiple strategies. We then train a router-generator factorization on samples from this fixed response distribution,

$$(x, y) \sim \mathcal{D} \quad \text{for base model fitting,} \qquad x \sim \mathcal{D}_X, \; y \sim p_\theta(\cdot|x) \quad \text{for factorization training.}$$

We report two primary metrics: *Distributional Fidelity*, whether the factorized model preserves the base model's behavior (i.e., still produces valid solution traces), and *Strategy Alignment*, whether the same latent value selects the same strategy across different problems (cross-input semantic consistency). Objective settings and run selection are in Appendix E.3.

4.2 Recovering Strategy-Aligned Latents

Implicit strategy information. We first ask whether the base model already contains information about the reference strategy before any router-generator factorization is trained. For each task and token position, we train a lightweight linear probe on the base model's hidden states to predict the reference strategy annotation. Figure 5 shows that strategy information is broadly decodable across the algorithmic benchmark, especially at positions where strategy-specific computation is expressed in the sequence. This provides evidence that the target strategy structure is represented in the base model's hidden states, but it is not exposed as a reusable latent variable that can be sampled, transferred, or compared across inputs.

ELBO failure. The presence of implicit strategy information does not imply that a standard variational objective will identify it. The conditional ELBO is under-specified for this pur-

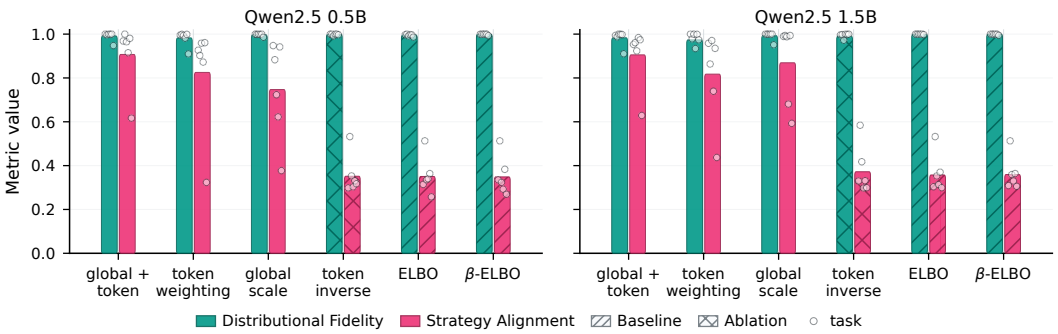


Figure 6: **Method comparison for Qwen2.5-0.5B and Qwen2.5-1.5B.** For each objective variant and model size, bars show Distributional Fidelity and Strategy Alignment, with open points giving per-task values. Strategy Alignment denotes cross-input latent strategy alignment measured by Analogical Consistency. Hatching marks baselines and ablations.

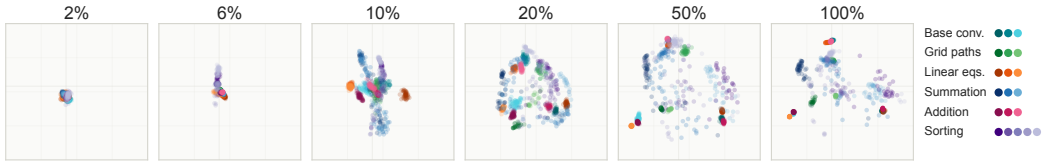


Figure 7: **Posterior latent geometry over training.** Each panel is a PCA projection of posterior mean latents at the indicated fraction of training. Points are colored by task and strategy; the initially overlapping cloud separates into task- and strategy-specific regions over training.

pose: it rewards matching the observable response distribution, but does not require z to explain strategy-level variation. Empirically, Figure 3 and Figure 6 show that ELBO-style baselines can preserve Distributional Fidelity while leaving the latent inert or strategy-unaligned.

The proposed model-directed reconstruction objective changes which marginal-preserving factorization is learned. Figure 6 shows results for the two Qwen2.5 models, and Figure 10 for SmolLM2-360M and OLMo 2 1B. Across all four base models, the model-directed objectives recover strategy-aligned latents with high Distributional Fidelity, whereas ELBO-style baselines preserve fidelity but not alignment. The latent dynamics in Figure 7 provide complementary geometric evidence: posterior latents begin as an overlapping cloud and separate into task- and strategy-structured regions over training. Together, these results indicate that model-directed reconstruction pressure can expose the strategy structure implicit in the base model's response distribution.

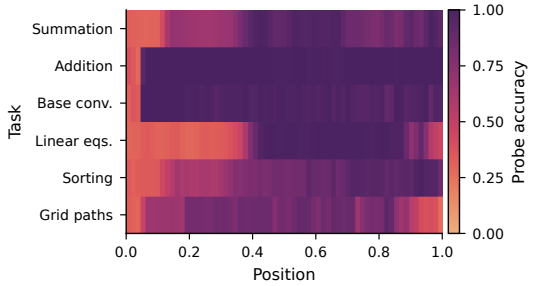


Figure 5: **Strategy labels are decodable from base model hidden states.** Linear-probe accuracy for the ground-truth strategy, by task (rows) and normalized token position (columns).

Cross-input semantic consistency. A latent variable can separate strategies locally for each input while failing to give Z a stable strategy meaning across inputs. We therefore evaluate a stronger notion of Strategy Alignment with Analogical Consistency: sample a source input X, draw $Z \sim r_\phi(\cdot|X)$, reuse the same Z on a related target input X', and check whether $Y \sim g_\phi(\cdot|X, Z)$ and $Y' \sim g_\phi(\cdot|X', Z)$ use the same inferred strategy. "Strategy Alignment" in the main figures denotes this cross-input metric, $\mathrm{AnalogicalConsistency} = \Pr[\mathrm{strat}(Y) = \mathrm{strat}(Y')]$, and Figure 6 shows that our objective achieves it. Analogical Consistency is thus an intervention experiment that measures whether setting the latent selects the generated strategy consistently across inputs.

Ablation: *inverse*-surprisal reconstruction pressure. The proposed objective puts more reconstruction pressure on high-surprisal tokens under p_θ, where strategy-branching information is expected to concentrate. We test this hypothesis through an ablation in which the reconstruction loss

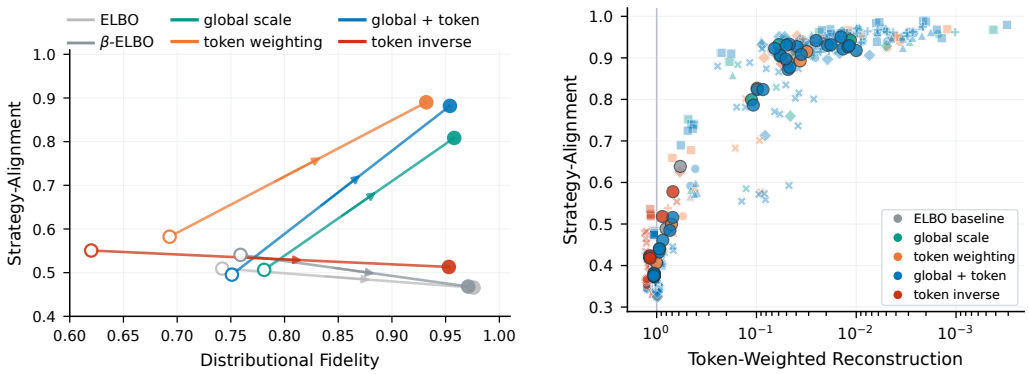


Figure 8: **Left:** training trajectories in fidelity-alignment phase space for different objective variants. **Right:** token-weighted reconstruction loss versus Strategy Alignment at training endpoints; outlined points show method centroids and translucent points show individual runs.

is weighted by *inverse* surprisal, so that low-surprisal tokens receive more emphasis. This ablation weakens Strategy Alignment, supporting the view that the model-directed reconstruction helps by pressuring z to encode strategy-relevant response variation. Details are in Appendix E.

How Model-Directed Reconstruction Pressure Drives Strategy Alignment. The preceding results show that the proposed objective recovers strategy-aligned latents. We next ask which part of the training signal explains the alignment gains across proposed methods and variational baselines. If alignment comes from capturing high-surprisal response variation in z, runs with lower token-weighted reconstruction loss should have higher Strategy Alignment.

Figure 8 supports this mechanism-level interpretation. The phase-space dynamics show that, when the objective is underspecified, the model can move toward high Distributional Fidelity without moving toward high Strategy Alignment. The scatter plot shows that token-weighted reconstruction loss tracks Strategy Alignment across all methods. This suggests strategy alignment improves when z is pressured to explain strategy-relevant residual surprisal under p_θ.

Efficient Strategy Exploration via Latent Space Intervention. The *analogical consistency* measure shows that the latent structure learned by our method enables strategy control via intervention in latent space. We now show that such intervention can be used as a mechanism for performing search and *active exploration in strategy space*. The goal is to discover more distinct strategies in the model's response distribution with fewer generations; that is, greater *exploration efficiency*. We consider a simple *latent exploration* method which biases latent sampling from the router towards under-explored regions of latent space. At each step $t = 1, \ldots, T$, where T is the sampling budget, it draws $k = 64$ candidate latents from the router, whitens them by the router's mean and covariance, and selects the candidate whose direction is farthest in angle from all previously selected latents (Algorithm 2). The generator then produces the response conditioned on these latents.

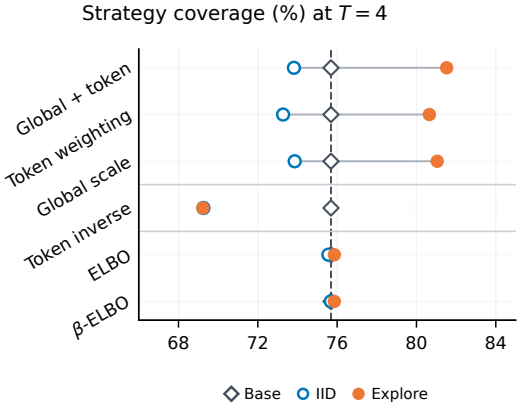


Figure 9: Intervention in latent space enables more efficient exploration of distinct strategies.

We evaluate the *coverage* of distinct strategies discovered through this process, compared to standard IID sampling from the base model (Figure 9; details in Appendix E.6). All three model-directed objectives improve exploration efficiency, while the ELBO baselines and the inverse-surprisal ablation do not. This further shows that the latent geometry learned by the model-directed objectives is aligned with the strategy variation in the response distribution, and demonstrates the promise of such methods for efficient active exploration.

5 Toward a Theory of Strategy-Structure Recovery

This section outlines a line of analysis that can provide theoretical insight into the methods introduced in the paper. The goal is to understand when the variational approach can identify a latent variable Z that aligns with a target strategy structure $S^\star$ implicit in the pretrained model. We provide a preview of the main results here; the full formal treatment is given in Appendix F.

Consider finite sets $\mathcal{X}$ and $\mathcal{Y}$, an input law μ, and a fixed pretrained response distribution $p(y \mid x)$. For an encoder $q(z \mid x, y)$, router $r(z \mid x)$, and generator $g(y \mid x, z)$, let Q_q be the joint law obtained by sampling $X \sim \mu$, $Y \sim p(\cdot \mid X)$, and $Z \sim q(\cdot \mid X, Y)$. We study, for $\beta \in [0, 1)$ and $\lambda \geq 0$, the population complexity-penalized variational objective

$$\mathcal{L}_{\beta,\lambda}(q, r, g) := \mathbb{E}_{Q_q}[-\log g(Y \mid X, Z)] + \beta\, \mathbb{E}_{X,Y} \mathrm{KL}(q(\cdot \mid X, Y) \| r(\cdot \mid X)) + \lambda C(r, g), \tag{5}$$

where $C(r, g)$ is a theoretical proxy for architecture, parameterization, and optimization biases that favor simple router-generator factorizations. The appendix shows the decomposition

$$\begin{aligned} \mathcal{L}_{\beta,\lambda}(q, r, g) = H(Y \mid X) - (1 - \beta) I_q(Y; Z \mid X) \\ + \mathrm{DecGap}(q, g) + \beta\, \mathrm{PriorGap}(q, r) + \lambda C(r, g), \end{aligned} \tag{6}$$

where DecGap and PriorGap are the generator and router KL gaps. For a fixed encoder q, define

$$A_{\beta,\lambda}(q) := \inf_{r \in \mathcal{R},\ g \in \mathcal{G}} \left[\mathrm{DecGap}(q, g) + \beta\, \mathrm{PriorGap}(q, r) + \lambda C(r, g)\right]. \tag{7}$$

We show that minimizing $\mathcal{L}_{\beta,\lambda}$ is equivalent to maximizing the following profiled score

$$\mathcal{J}^{\mathrm{prof}}_{\beta,\lambda}(q) := I_q(Y; Z \mid X) - \frac{A_{\beta,\lambda}(q)}{1 - \beta}. \tag{8}$$

The first term $I_q(Y; Z \mid X)$ is the information pressure that encourages Z to capture information about Y beyond what is already predictable from X, while the second term $A_{\beta,\lambda}(q)$ is an effective approximation-complexity penalty that measures whether the induced distribution is close to the target while being simple, as measured by the complexity term $C(r, g)$.

Let $S^\star = f(X, Y)$ be the targeted strategy, and let $q^\star(z|x, y) = \mathbf{1}\{z = S^\star\}$. Let $d(q, S^\star)$ be a recovery discrepancy between q and $S^\star$. Our primary result gives sufficient conditions under which the complexity-penalized variational objective selects a latent factorization aligned with $S^\star$.

Theorem (informal). *Define* $\Delta^{\mathrm{prof}}_{\beta,\lambda}(\delta) := \mathcal{J}^{\mathrm{prof}}_{\beta,\lambda}(q^\star) - \sup_{q \in \mathcal{Q}:\, d(q,S^\star) \geq \delta} \mathcal{J}^{\mathrm{prof}}_{\beta,\lambda}(q)$*, for some discrepancy measure* $d(q, S^\star)$ *and assume* $\Delta > 0$*. Then, given* $n \gtrsim \log(|\mathcal{Q}||\mathcal{R}||\mathcal{G}|/\eta)/((1 - \beta)^2 \Delta^2)$ *samples, the empirical minimizer of eq. (6) recovers* $S^\star$ *up to distance* δ *with probability* $1 - \eta$*.*

6 Conclusion

This paper studies how to recover explicit latent representations of reasoning strategies from pretrained language models. Because the base generator already realizes the response distribution, a factorization that ignores the latent is globally optimal under objectives that only measure marginal fit, and we show that standard variational training can leave the latent unused. We propose a model-directed reconstruction objective that normalizes reconstruction by the base model's response loss and weights tokens by base model surprisal, concentrating pressure where strategy choices are made. On a controlled benchmark of multi-strategy algorithmic tasks, this objective recovers latents that are aligned with reference strategies and consistent across inputs, and intervening in this latent space enables more efficient exploration of the model's strategy space.

Our benchmark makes strategy recovery measurable but limits the scope of our conclusions: we establish recovery where reference strategies are enumerable and traces are parseable, not in open-ended reasoning, where strategies are not known in advance, may overlap or be hierarchical, and where response variation also reflects style, formatting, and other non-strategy factors. Extending to open-ended settings requires evaluation that does not rely on a parser, for example model-based judgments of whether two responses instantiate the same strategy. Moreover, the theoretical analysis is preliminary, and the connection between the oracle selection criterion and actual training dynamics is not yet formally characterized. Appendix A discusses related work and future directions.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, including the training code and the synthetic task generators, and to assist in writing proofs. Additionally, we used generative AI tools to create and modify figures, search for information, identify and summarize relevant literature, and edit the paper for readability. We have not used generative AI tools to design research methodology, generate synthetic data sets, or clean and reformat data sets. All AI-generated code was reviewed and tested by the authors, and all proofs were verified by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Allal, Loubna Ben et al. (2025). "SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model". arXiv: 2502.02737 [cs.CL] (cited on page 20).

Allman, Elizabeth S., Catherine Matias, and John A. Rhodes (2009). "Identifiability of Parameters in Latent Structure Models with Many Observed Variables". In: *The Annals of Statistics* (cited on pages 13, 14).

Bao, Yu, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen (2019). "Generating Sentences from Disentangled Syntactic and Semantic Spaces". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (cited on pages 13, 14).

Bowman, Samuel R., Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio (2016). "Generating Sentences from a Continuous Space". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (cited on pages 1, 13, 14).

Burgess, Christopher P., Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner (2018). "Understanding disentangling in β-VAE". arXiv: 1804.03599 [stat.ML] (cited on pages 13, 14).

Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel (2016). "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems* (cited on pages 13, 14).

Fu, Hao, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin (2019). "Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (cited on pages 1, 14).

Galichin, Andrey, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan Oseledets (2025). "I Have Covered All the Bases Here: Interpreting Reasoning Features in Large Language Models via Sparse Autoencoders". arXiv: 2503.18878 [cs.CL] (cited on pages 13, 15).

Gu, Yuxuan, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, Weihong Zhong, and Bing Qin (2023). "Controllable Text Generation via Probability Density Estimation in the Latent Space". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (cited on page 14).

He, Junxian, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick (2019). "Lagging Inference Networks and Posterior Collapse in Variational Autoencoders". In: *International Conference on Learning Representations* (cited on pages 1, 14).

Higgins, Irina, Loic Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner (2017). "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations* (cited on pages 13, 14).

Hoffman, Matthew D. and Matthew J. Johnson (2016). "ELBO Surgery: Yet Another Way to Carve up the Variational Evidence Lower Bound" (cited on pages 1, 14).

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen (2022). "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations* (cited on page 4).

Hyvärinen, Aapo, Hiroaki Sasaki, and Richard E. Turner (2019). "Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning". In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics* (cited on pages 13, 14).

Jiang, Wenxin and Martin A. Tanner (1999). "Hierarchical Mixtures-of-Experts for Exponential Family Regression Models: Approximation and Maximum Likelihood Estimation". In: *The Annals of Statistics* (cited on pages 13, 14).

Keskar, Nitish Shirish, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher (2019). "CTRL: A Conditional Transformer Language Model for Controllable Generation". arXiv: 1909.05858 [cs.CL] (cited on page 2).

Khemakhem, Ilyes, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen (2020). "Variational Autoencoders and Nonlinear ICA: A Unifying Framework". In: *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics* (cited on pages 13, 14).

Kim, Hyunjik and Andriy Mnih (2018). "Disentangling by Factorising". In: *Proceedings of the 35th International Conference on Machine Learning* (cited on pages 13, 14).

Kingma, Diederik P. and Max Welling (2013). "Auto-Encoding Variational Bayes". arXiv: 1312.6114 [stat.ML] (cited on pages 1, 13, 14).

Koh, Pang Wei, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang (2020). "Concept Bottleneck Models". In: *Proceedings of the 37th International Conference on Machine Learning* (cited on page 2).

Kong, Deqian et al. (2025). "Latent Thought Models with Variational Bayes Inference-Time Computation". In: *Proceedings of the 42nd International Conference on Machine Learning* (cited on page 14).

Li, Chunyuan, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao (2020). "Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics (cited on pages 13, 14).

Li, Xiang Lisa and Percy Liang (2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (cited on page 2).

Mercatali, Giangiacomo and André Freitas (2021). "Disentangling Generative Factors in Natural Language with Discrete Variational Autoencoders". In: *Findings of the Association for Computational Linguistics: EMNLP 2021* (cited on pages 13, 14).

OLMo, Team et al. (2025). "2 OLMo 2 Furious". arXiv: 2501.00656 [cs.CL] (cited on page 20).

Park, Seongmin and Jihwa Lee (2021). "Finetuning Pretrained Transformers into Variational Autoencoders". In: *Proceedings of the Second Workshop on Insights from Negative Results in NLP* (cited on pages 13, 14).

Qwen Team (2024). "Qwen2.5 Technical Report". arXiv: 2412.15115 [cs.CL] (cited on page 20).

Serban, Iulian V., Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio (2017). "A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues". In: *Proceedings of the AAAI Conference on Artificial Intelligence* (cited on page 14).

Shazeer, Noam, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean (2017). "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". In: *International Conference on Learning Representations* (cited on page 13).

Sohn, Kihyuk, Honglak Lee, and Xinchen Yan (2015). "Learning Structured Output Representation using Deep Conditional Generative Models". In: *Advances in Neural Information Processing Systems* (cited on pages 1, 13, 14).

Tan, Zhen, Lu Cheng, Song Wang, Bo Yuan, Jundong Li, and Huan Liu (2024). "Interpreting Pretrained Language Models via Concept Bottlenecks". In: *Advances in Knowledge Discovery and Data Mining* (cited on pages 13, 15).

Teicher, Henry (1963). "Identifiability of Finite Mixtures". In: *The Annals of Mathematical Statistics* (cited on pages 13, 14).

Tuyls, Jens, Dylan J. Foster, Akshay Krishnamurthy, and Jordan T. Ash (2025). "Representation-Based Exploration for Language Models: From Test-Time to Post-Training". arXiv: 2510.11686 [cs.LG] (cited on pages 2, 15).

Vasilakes, Jake, Chrysoula Zerva, Makoto Miwa, and Sophia Ananiadou (2022). "Learning Disentangled Representations of Negation and Uncertainty". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (cited on pages 13, 14).

Wang, Xuezhi, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou (2023). "Self-Consistency Improves Chain of Thought Reasoning in Language Models". In: *International Conference on Learning Representations* (cited on page 2).

Wu, Fang, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi (2026). "The Invisible Leash: Why RLVR May or May Not Escape Its Origin". arXiv: 2507.14843 [cs.LG] (cited on page 15).

Yakowitz, Sidney and John D. Spragins (1968). "On the Identifiability of Finite Mixtures". In: *The Annals of Mathematical Statistics* (cited on pages 13, 14).

Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan (2023). "Tree of Thoughts: Deliberate Problem Solving with Large Language Models". In: *Advances in Neural Information Processing Systems* (cited on page 2).

Yue, Yang, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang (2025). "Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?" arXiv: 2504.13837 [cs.AI] (cited on pages 2, 15).

Zhang, Yingji, Marco Valentino, Danilo Carvalho, Ian Pratt-Hartmann, and Andre Freitas (2024). "Graph-Induced Syntactic-Semantic Spaces in Transformer-Based Variational AutoEncoders". In: *Findings of the Association for Computational Linguistics: NAACL 2024* (cited on pages 13, 14).

Zhao, Shengjia, Jiaming Song, and Stefano Ermon (2019). "InfoVAE: Balancing Learning and Inference in Variational Autoencoders". In: *Proceedings of the AAAI Conference on Artificial Intelligence* (cited on page 14).

Zhao, Tiancheng, Ran Zhao, and Maxine Eskenazi (2017). "Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (cited on pages 13, 14).

A Extended Discussion

This paper studies the problem of recovering explicit latent variable representations of reasoning strategies from pretrained language models. The central observation is that a pretrained model's response distribution is an entangled mixture over high-level strategies, and that exposing this structure requires more than fitting a latent variable model to observable outputs using previously explored methods. In particular, because the base generator already realizes this response distribution, a factorization that ignores the latent variable is immediately available and is globally optimal under objectives that only measure marginal fit.

We show that standard variational training can preserve the observable response distribution while leaving the latent variable unused. To address this, we propose a model-directed reconstruction objective that uses the base model itself as a reference. In our approach, reconstruction is normalized by the base model's response loss and weighted by the information content of each response token under the base model, allowing the loss to concentrate on positions where strategy choices are most consequential. On a controlled benchmark of multi-strategy algorithmic tasks, this objective recovers latent structure that is well-aligned with reference solution strategies, while also being semantically consistent across inputs. In this way, the latent variable transfers the same strategy from one problem to another.

A.1 Related Work

There is a rich and extensive literature on latent variable methods for generative models of text. This work includes VAEs (Kingma and Welling, 2013) and CVAEs (Sohn et al., 2015), text and sequence VAEs (Bowman et al., 2016; T. Zhao et al., 2017), and disentanglement-oriented variants such as the β-VAE (Higgins et al., 2017; Burgess et al., 2018), FactorVAE (Kim and Mnih, 2018), and InfoGAN (Chen et al., 2016), as well as pretrained-Transformer VAEs such as OPTIMUS (C. Li et al., 2020) and the Transformer VAE of Park and Lee (2021). Adjacent language-VAE work also uses structured latent spaces to separate syntax from semantics, represent discrete linguistic factors, and control semantic attributes such as negation or uncertainty (Bao et al., 2019; Mercatali and Freitas, 2021; Vasilakes et al., 2022; Zhang et al., 2024). The problem we study in this work is not directly addressed by this prior work. The key distinction is that these methods learn a latent variable model where the target distribution is not represented by the model at initialization. In our setting the base model already realizes $p_\theta(y|x)$, and it would be inefficient, if not impossible, to retrain the model from scratch to expose the strategy.

Mixture-of-experts language models also learn routing over specialized generators (Shazeer et al., 2017; Jiang and Tanner, 1999), but such models are optimized for capacity or efficiency, not for correspondence to an interpretable strategy variable, and training is typically carried out from scratch. Mechanistic interpretability methods, including sparse autoencoders (Galichin et al., 2025) and concept-bottleneck approaches (Tan et al., 2024), extract or steer internal representations post-hoc; our goal is instead a generative factorization that supports sampling and intervention.

While there is also a rich theory for latent variable methods, classical identifiability theory for mixture models (Teicher, 1963; Yakowitz and Spragins, 1968; Allman et al., 2009) and nonlinear ICA (Hyvärinen et al., 2019; Khemakhem et al., 2020) achieve uniqueness by restricting the component family or adding auxiliary observations. When using an expressive pretrained generator, classical identifiability conditions generally fail, and the central question becomes how to select a useful factorization among the many that preserve the marginal distribution. An extended discussion of related work with fuller citations is given in the supplementary material.

A.2 Implications and Future Directions

The ideas and methods presented here open up several directions for future work; here we briefly mention potential connections to interpretability, broader directed exploration, and safety. First, because a strategy-aligned factorization exposes the strategies implicit in a pretrained model's response distribution and makes them comparable across related inputs, it can provide a higher-level and more interpretable description of model behavior than token-level analysis. Moreover, intervening on z at generation time can direct the model toward a specific strategy; shifting the router distribution $r_\phi(z \mid x)$ adjusts the relative weight of different strategies without retraining. This has the potential to provide a structured prior for exploring the strategy space, which could improve coverage in settings where the base model under-samples certain valid approaches. Finally, because strategy-level control offers a more interpretable abstraction for constraining or auditing generation compared with token-level steering, the methods introduced in this paper may support applications where certain reasoning paths need to be permitted or excluded, presenting possible benefits for safety.

The scope of our conclusions is limited in two respects. Our evaluation establishes strategy recovery where reference strategies are enumerable and traces are parseable. It does not establish recovery in open-ended reasoning, where strategies are not known in advance, may overlap or be hierarchical, and where response variation also reflects style, formatting, and other factors that are not strategy. Our directed-exploration experiment establishes improved strategy coverage on this controlled benchmark, but does not evaluate applications in reinforcement-learning post-training. Extending to open-ended settings requires evaluation that does not rely on a parser, for example model-based judgments of whether two responses instantiate the same strategy. Moreover, the theoretical analysis is preliminary, and the connection between the oracle selection criterion and actual training dynamics is not yet formally characterized.

B Extended Related Work

Conditional latent-variable models and language VAEs. Latent-variable generative models provide the formal background for our router-generator factorization. The VAE and conditional VAE literature learns a latent-variable model from observed data, using an encoder or recognition model and an ELBO to train a generator with latent variation (Kingma and Welling, 2013; Sohn et al., 2015). Sequence and dialogue VAEs extend this idea to autoregressive generation, where the latent is intended to capture sentence-level, discourse-level, or response-level variation beyond the input context (Bowman et al., 2016; Serban et al., 2017; T. Zhao et al., 2017). Pretrained language VAEs and latent-thought models scale related ideas with pretrained Transformer components, continuous sentence spaces, or inference-time latent optimization (C. Li et al., 2020; Park and Lee, 2021; Kong et al., 2025). Another line of language-VAE work explicitly structures the latent space around controllable or disentangled linguistic factors. Prior work separates syntax and semantics in sentence VAEs (Bao et al., 2019), extends syntactic-semantic separation to Transformer-based VAEs (Zhang et al., 2024), uses discrete latent variables to capture natural-language generative factors (Mercatali and Freitas, 2021), and disentangles semantic attributes such as negation and uncertainty from content (Vasilakes et al., 2022). Latent-space control methods further model control distributions directly in the VAE latent space (Gu et al., 2023). Our setting differs in the learning target: rather than fitting a latent-variable generator to data, we start from a fixed or pretrained response distribution $p_\theta(y|x)$ and learn a factorization $\int r_\phi(z|x)g_\phi(y|x,z)\,dz$ whose latent variable exposes strategy-level variation already implicit in that distribution.

Posterior collapse, latent usage, and disentanglement objectives. Autoregressive decoders can ignore latent variables, giving rise to posterior-collapse and KL-vanishing pathologies in sequence VAEs (Bowman et al., 2016; He et al., 2019; Fu et al., 2019). Several objective analyses separate the standard ELBO regularizer into latent information and aggregate-prior matching terms, motivating variants that preserve latent usage while keeping generation well regularized (Hoffman and Johnson, 2016; S. Zhao et al., 2019). Disentanglement-oriented objectives such as β-VAE, capacity-controlled β-VAE, FactorVAE, and InfoGAN also show that useful latent structure usually requires explicit inductive bias or information pressure, not merely the presence of a latent code (Higgins et al., 2017; Burgess et al., 2018; Kim and Mnih, 2018; Chen et al., 2016). These works motivate our use of objective terms that pressure z to carry information. However, our desideratum is more specific than latent activity or visual-factor disentanglement: z should be informative about y beyond x, aligned with reference solution strategies, and semantically consistent across related inputs. The collapse issue is also sharper in our adaptation setting because the initialized generator already realizes the observable conditional behavior without consulting z.

Identifiability, conditional mixtures, and routing. Classical finite-mixture theory studies when a marginal distribution uniquely determines its mixture components, often through restrictive component families, linear-independence conditions, or multi-view structure (Teicher, 1963; Yakowitz and Spragins, 1968; Allman et al., 2009). Conditional mixture and mixture-of-experts models introduce input-dependent routing or gating, but their classical guarantees usually concern approximation, prediction, or consistency under structured expert families rather than recovery of a semantically named latent variable (Jiang and Tanner, 1999). Identifiable latent-variable and nonlinear-ICA results restore stronger recovery guarantees by adding auxiliary variables or conditional-prior structure that breaks otherwise unavoidable symmetries (Hyvärinen et al., 2019; Khemakhem et al., 2020). These literatures clarify why latent-variable factorization is not identified by marginal fit alone. In our setting, the router and generator are expressive neural language models, so many factorizations can preserve $p_\theta(y|x)$. The problem is therefore not only estimation of the observable conditional law, but selection of a useful factorization whose latent variable captures reusable strategy structure.

Interpretability and reasoning features. A separate line of work exposes internal model structure through supervised concept bottlenecks or post-hoc feature decompositions. Concept-bottleneck methods for pretrained language models insert human-defined concept variables and train the model to predict through them, providing an explicit supervised interpretability interface (Tan et al., 2024). Sparse-autoencoder analyses of reasoning models instead discover internal features associated with reasoning motifs and can steer model behavior by intervening on feature directions (Galichin et al., 2025). These approaches are closely related to our interpretability motivation because they aim to make latent reasoning structure visible and manipulable. The distinction is that we learn a generative router-generator factorization of $p_\theta(y|x)$, so the learned latent variable participates directly in generation rather than serving only as a supervised concept layer or a post-hoc activation feature.

Exploration and RLVR motivation. Recent analyses of reinforcement learning with verifiable rewards suggest that on-policy post-training often sharpens solutions already accessible under the base model rather than broadly expanding the set of reachable reasoning strategies (Yue et al., 2025; Wu et al., 2026). Representation-based exploration methods respond to this limitation by adding novelty or diversity pressure in model representation space (Tuyls et al., 2025). This literature motivates strategy-level intervention: a reusable strategy-aligned z could provide a higher-level handle for directed exploration than token-space sampling alone. Our controlled-benchmark experiment measures strategy coverage under passive and directed latent sampling. We do not evaluate RLVR or exploration in open-ended reasoning settings.

C Benchmark and Data Generation

The controlled multi-strategy algorithmic benchmark provides problem instances, solution traces, and reference strategy labels in a setting where strategy alignment can be measured directly. Each task family is designed so that the same problem can be solved by multiple valid procedures. The procedures differ in the intermediate solution trace while preserving the same final answer. This gives an evaluation reference for asking whether a learned latent variable corresponds to strategy-level variation in generated reasoning traces.

C.1 Task families

Table 1 summarizes the benchmark tasks used in the paper. Within each task family, strategies are sampled uniformly from the listed strategy set. The table lists the sampled strategies, not every procedure that could in principle solve the task. The task families probe complementary forms of strategy variation: reduction order for summation, state-transition algorithms for sorting, move-order policies for grid paths, algebraic manipulation order for linear equations, conversion procedures for base conversion, and arithmetic decomposition for multidigit addition. In each case, the strategies alter the intermediate trace while preserving the same problem-answer relation.

C.2 Data generation

The benchmark data distribution samples a task family, a problem instance for that task, and a reference strategy for solving that instance:

$$T \sim \mathrm{Unif}(\text{task families}), \qquad X \sim \mathcal{D}_T, \qquad S \sim \mathrm{Unif}(\mathcal{S}_T), \qquad Y = \mathrm{Trace}_T(X, S).$$

Here $\mathcal{D}_T$ is the task-specific input distribution and $\mathcal{S}_T$ is the listed strategy set for task T. The rendering function Trace_T produces a solution trace that follows strategy S and reaches the task's correct final answer. The resulting example consists of a problem instance X, a solution trace Y, and a reference strategy label S that is used for evaluation but not training.

C.3 Reference strategy identification

The benchmark is designed to enable parsing solution trajectories generated by a model and identifying the reference strategy. Given a problem instance and a generated solution trace, the task parser checks whether the trace is a valid solution and identifies which benchmark strategy it expresses. This makes it possible to evaluate whether generated outputs preserve valid task behavior and whether a learned latent variable is aligned with the benchmark's reference strategy structure. The reference labels are used as an evaluation target for the controlled benchmark.

C.4 Representative examples

To illustrate the benchmark, the examples below show a sample problem from each task with two sample solutions demonstrating two different strategies. Some traces are abbreviated with ellipses where the omitted steps continue the same procedure.

Table 1: **Controlled multi-strategy algorithmic benchmark.** Each task family admits multiple valid solution procedures for the same problem instance. The benchmark uses the listed sampled strategies as reference labels for evaluating strategy-aligned latent variables.

Task family	Problem domain	Sampled strategies
List summation	Four integers from 0 to 9	left-to-right; right-to-left; pairwise
Sorting algorithms	Five integers from 0 to 9	bubble sort; insertion sort; merge sort
Grid pathfinding	Monotone shortest paths on a 6×6 grid	right-first; down-first; alternating
Linear equation solving	Integer equations $ax + b = c$, with $a = 2, \ldots, 9$ and integer-safe solutions and offsets in $[-9, 9]$	subtract-then-divide; divide-then-subtract; inverse-ops
Base conversion	Integer $n = 1, \ldots, 255$, base in $2, 4, 8, 16$	repeated division; via binary; decomposition
Multidigit addition	Two three-digit nonnegative integers	right-to-left carry; left-to-right partials; rounding decomposition

List summation. For the problem instance "`3, 7, 2, 8`", the benchmark can render traces such as:

Strategy	Trace
left-to-right	`3+7=10 ; 10+2=12 ; 12+8=20`
pairwise	`3+7=10 ; 2+8=10 ; 10+10=20`

Sorting algorithms. For the problem instance "`[5,2,8,1,4]`", the benchmark can render traces such as:

Strategy	Trace
bubble sort	`[2,5,8,1,4] ; [2,5,1,8,4] ; ... ; [1,2,4,5,8]`
merge sort	`[2,5,8,1,4] ; [2,5,1,8,4] ; [1,2,5,8,4] ; [1,2,4,5,8]`

Grid pathfinding. For the problem instance "`grid=6 ; start=(0,0) ; goal=(2,3)`", the benchmark can render traces such as:

Strategy	Trace
right-first	`R:(0,1) ; R:(0,2) ; R:(0,3) ; D:(1,3) ; D:(2,3)`
alternating	`R:(0,1) ; D:(1,1) ; R:(1,2) ; D:(2,2) ; R:(2,3)`

Linear equation solving. For the problem instance "`3x+6=21`", the benchmark can render traces such as:

Strategy	Trace
subtract-then-divide	`3x+6=21 ; 3x=15 ; x=5`
divide-then-subtract	`3x+6=21 ; x+2=7 ; x=7-2 ; x=5`

Base conversion. For the problem instance "`n=45, base=4`", the benchmark can render traces such as:

Algorithm 1: Model-directed training of a router-generator factorization

Input: Input distribution $\mathcal{D}_X$; frozen base model p_θ; objective parameters β, α, γ.
Output: Router $r_\phi(z|x)$ and strategy-conditioned generator $g_\phi(y|x, z)$.

Initialize ϕ from θ, adding router heads and a latent embedding projection
Initialize ξ from θ, adding posterior heads
Freeze p_θ as both the response distribution to factorize and the reconstruction reference
Estimate $c_\theta = \mathbb{E}_{x\sim\mathcal{D}_X,\, y\sim p_\theta(\cdot|x)}[b_\theta(x, y)]$ using frozen-base samples
for *each optimization step* **do**
    Sample a minibatch $x_i \sim \mathcal{D}_X$, for $i = 1, \ldots, B$
    Sample responses $y_i \sim p_\theta(\cdot|x_i)$
    Compute frozen-base surprisals $b_{\theta,t}(x_i, y_i)$ and weights $a_{\theta,t}(x_i, y_i)$
    Sample or reparameterize $z_i \sim q_\xi(\cdot|x_i, y_i)$
    Form $\widehat{\mathcal{J}}_\theta = B^{-1} \sum_{i=1}^{B} \mathcal{J}_\theta(x_i, y_i; \phi, \xi)$
    Update ϕ and ξ by a stochastic gradient step on $\widehat{\mathcal{J}}_\theta$, leaving θ fixed
end

Strategy	Trace
repeated division	`DIV:45/4=11,r:1 ; DIV:11/4=2,r:3 ; DIV:2/4=0,r:2 ; OUT:231`
decomposition	`TERM:2*4^2=32 ; TERM:3*4^1=12 ; TERM:1*4^0=1 ; OUT:231`

Multidigit addition. For the problem instance "`348+274`", the benchmark can render traces such as:

Strategy	Trace
right-to-left carry	`d2:8+4+0=12->w2,c1 ; d1:4+7+1=12->w2,c1 ; d0:3+2+1=6->w6,c0 ; sum=622`
left-to-right partials	`p100:300+200=500 ; p10:40+70=110 ; p1:8+4=12 ; sum=500+110+12=622`

D Details of Training Methodology

Section 3 describes the objective-level motivation for model-directed reconstruction. This appendix gives the corresponding training procedure and the implementation details needed to reproduce the factorization objective. Throughout this section, p_θ is a frozen base model, ϕ denotes the adapted router-generator parameters, and ξ denotes the training-time posterior parameters. The training distribution is generated by the frozen base model itself: inputs are drawn from $\mathcal{D}_X$ and responses are sampled from $p_\theta(\cdot|x)$.

D.1 Architecture Parameterization

The router and generator are implemented as two roles of one adapted autoregressive Transformer. The router reads the input prefix and produces a Gaussian distribution over continuous latent values. Let $h_\phi^{(L)}(x)$ be the final-layer representation produced by the adapted Transformer after reading x. The router distribution is

$$r_\phi(z|x) = \mathcal{N}\Big(W_\mu h_\phi^{(L)}(x), \operatorname{diag}\Big(\exp(W_\sigma h_\phi^{(L)}(x))\Big)\Big).$$

During autoregressive generation, a latent sample z is projected to an embedding $E_\phi(z)$ and inserted as a pseudo-token at the embedding layer. The generator then predicts each response token while attending to x, the latent pseudo-token, and the previous response tokens.

The training-time posterior uses the same construction, but reads the full input-response sequence. Let $h_\xi^{(L)}(x, y)$ be the final-layer representation after reading (x, y). The posterior distribution is

$$q_\xi(z|x, y) = \mathcal{N}\Big(U_\mu h_\xi^{(L)}(x, y), \operatorname{diag}\Big(\exp(U_\sigma h_\xi^{(L)}(x, y))\Big)\Big).$$

The posterior is used only during training to provide variational signal. At generation time, latents are sampled from the router.

Both adapted models are initialized from the frozen base model parameters θ. In the experiments, the trainable parameters consist of LoRA-style low-rank updates to the Transformer backbone together with the Gaussian heads and latent embedding projection. This keeps the factorization close to the base model while allowing the router and generator to assign a nontrivial role to z.

D.2 Objective and Normalization Details

For a sampled response $y = (y_1, \ldots, y_{T_y})$, define the frozen-base token surprisal and response-average surprisal as

$$b_{\theta,t}(x, y) = -\log p_\theta(y_t|x, y_{<t}), \qquad b_\theta(x, y) = \frac{1}{T_y} \sum_{t=1}^{T_y} b_{\theta,t}(x, y).$$

The global reference scale is

$$c_\theta = \mathbb{E}_{x\sim\mathcal{D}_X,\, y\sim p_\theta(\cdot|x)} \left[b_\theta(x, y) \right].$$

The θ subscript marks dependence on the frozen base model.

Token-level reconstruction weights are derived from the base model surprisals. For $\gamma \geq 0$, define

$$\widetilde{w}^{(\gamma)}_{\theta,t}(x, y) = \frac{b_{\theta,t}(x, y)^\gamma}{\sum_{s=1}^{T_y} b_{\theta,s}(x, y)^\gamma}, \qquad \kappa^{(\gamma)}_\theta(x, y) = \frac{\sum_{s=1}^{T_y} \widetilde{w}^{(\gamma)}_{\theta,s}(x, y) b_{\theta,s}(x, y)}{b_\theta(x, y)},$$

and

$$w^{(\gamma)}_{\theta,t}(x, y) = \frac{\widetilde{w}^{(\gamma)}_{\theta,t}(x, y)}{\kappa^{(\gamma)}_\theta(x, y)}, \qquad a_{\theta,t}(x, y) = \alpha \frac{1}{T_y} + (1 - \alpha)\, w^{(\gamma)}_{\theta,t}(x, y).$$

The parameter $\alpha \in [0, 1]$ mixes uniform response-token pressure with token-level p_θ-surprisal-directed pressure. The parameter γ controls how strongly the token weights concentrate on high-surprisal positions. The base model reference scores, token weights, and normalization constants defined above are treated as constants when updating ϕ and ξ.

The resulting reconstruction term and training objective are

$$\mathcal{R}_\theta(x, y; \phi, \xi) = \frac{1}{c_\theta} \mathop{\mathbb{E}}_{z\sim q_\xi(\cdot|x,y)} \left[\sum_{t=1}^{T_y} a_{\theta,t}(x, y) \left[-\log g_\phi(y_t|x, z, y_{<t}) \right] \right],$$

and

$$\mathcal{J}_\theta(x, y; \phi, \xi) = \mathcal{R}_\theta(x, y; \phi, \xi) + \beta\, \mathrm{KL}(q_\xi(\cdot|x, y) \,\|\, r_\phi(\cdot|x)).$$

The normalization by $\kappa^{(\gamma)}_\theta$ preserves the frozen-base response scale under token weighting:

$$\sum_{t=1}^{T_y} w^{(\gamma)}_{\theta,t}(x, y)\, b_{\theta,t}(x, y) = b_\theta(x, y).$$

Since the uniform component also satisfies

$$\sum_{t=1}^{T_y} \frac{1}{T_y} b_{\theta,t}(x, y) = b_\theta(x, y),$$

the mixed weights satisfy

$$\sum_{t=1}^{T_y} a_{\theta,t}(x, y)\, b_{\theta,t}(x, y) = b_\theta(x, y).$$

Consequently, if the generator matches the frozen base model token distribution, so that $g_\phi(y_t|x, z, y_{<t}) = p_\theta(y_t|x, y_{<t})$, then $\mathcal{R}_\theta(x, y; \phi, \xi) = b_\theta(x, y)/c_\theta$. Taking expectation over $x \sim \mathcal{D}_X$ and $y \sim p_\theta(\cdot|x)$ gives $\mathbb{E}[\mathcal{R}_\theta] = 1$. Thus $\mathcal{R}_\theta$ can be read as a base-normalized fraction of the frozen-base response loss that remains after conditioning on the latent.

D.3 Inference And Evaluation Interface

The posterior $q_\xi(z|x, y)$ is a training-time object. It provides latent samples for reconstruction and a variational bridge between the generator and router, but it is not needed to sample from the learned factorization. At inference time, the model first samples $z \sim r_\phi(\cdot|x)$ and then generates y autoregressively from $g_\phi(\cdot|x, z)$.

Strategy annotations are not used in the optimization loop. They enter only after training, when we evaluate whether the learned latent is informative and strategy-aligned, separates strategies within each input, and assigns semantically consistent latent regions across inputs

E Experimental Details

E.1 Evaluation protocol and metrics

We evaluate trained router-generator factorizations using the parseable structure of the benchmark described in Appendix C. Generated solution traces can be parsed and assigned to the benchmark's reference strategy structure, enabling direct evaluation of whether the learned latent variable preserves task behavior and has stable strategy meaning across related inputs. Strategy labels are used only for evaluation.

Distributional Fidelity. Distributional Fidelity measures whether samples from the factorized model preserve the base model response behavior at the level of valid benchmark solutions. Operationally, we sample $Z \sim r_\phi(\cdot|X)$, generate $Y \sim g_\phi(\cdot|X, Z)$, parse the generated trace, and report the fraction of samples that remain strategy-compatible task solutions.

Strategy Alignment. The main-paper Strategy Alignment metric is Analogical Consistency, as defined in Section 4.2. We sample a source input X, draw $Z \sim r_\phi(\cdot|X)$, generate $Y \sim g_\phi(\cdot|X, Z)$, reuse the same Z on a related target input X', and generate $Y' \sim g_\phi(\cdot|X', Z)$. The score is

$$\text{AnalogicalConsistency} = \Pr[\text{strat}(Y) = \text{strat}(Y')]\,,$$

where X' is sampled from the same task family as X. This tests whether the latent value carries the same strategy meaning across different inputs.

Supporting diagnostics. Router Strategy Decodability is a linear-probe diagnostic measuring whether reference strategy information is linearly recoverable from router-sampled latents $Z \sim r_\phi(\cdot|X)$. It is useful for diagnosing latent information, but it is weaker than Analogical Consistency because it does not test whether the same latent value has reusable cross-input meaning. For mechanism analysis, we also report base-relative reconstruction $\mathcal{R}_\theta$, token-weighted reconstruction, and the posterior-router prior KL term from the training objective.

E.2 Objective variants and inverse-surprisal ablation

Appendix D.2 gives the full model-directed objective. The experiments compare variants of

$$\mathcal{J}(x, y; \phi, \xi) = \mathcal{R}(x, y; \phi, \xi) + \beta\,\mathrm{KL}(q_\xi(\cdot|x, y)\,\|\,r_\phi(\cdot|x))\,,$$

where the variants differ in the reconstruction term $\mathcal{R}$. The standard ELBO uses the unnormalized autoregressive reconstruction loss, while the proposed variants use the frozen base model p_θ to normalize and direct reconstruction pressure.

For uniform base-relative reconstruction, the token weights are $a_{\theta,t} = 1/T_y$. For surprisal-proportionate token weighting, $a_{\theta,t}$ is the concentration-corrected token weight defined from the frozen-base surprisal $b_{\theta,t}(x, y)$ in Appendix D.2. The global+token variants use convex mixtures of the uniform base-relative term and the high-surprisal token-weighted term. Table 2 maps the paper-facing method labels to the objective components used in the figures.

Table 2: **Objective variants used in the experimental figures.** All model-directed variants train on samples $x \sim \mathcal{D}_X,\ y \sim p_\theta(\cdot|x)$ and keep p_θ frozen as the reference model.

Method label	Reconstruction term	Role in comparison
ELBO baseline	Standard token reconstruction, without base-relative normalization	Tests whether ordinary variational training identifies strategy-aligned latents.
Global scale	Base-relative reconstruction with uniform token weights	Tests the effect of putting reconstruction in frozen-base reference units.
Token weighting	Base-relative reconstruction with high-surprisal token weights	Tests model-directed pressure toward response regions where p_θ has residual surprisal.
Global + token	Convex mixture of uniform base-relative and high-surprisal token-weighted reconstruction	Combines global normalization with token-level model-directed pressure.
Token inverse	Base-relative reconstruction with inverse-surprisal token weights	Directional negative control emphasizing low-surprisal tokens.

Inverse-surprisal ablation. The inverse-surprisal ablation tests whether the direction of token-level pressure matters. Let $\tilde{b}_{\theta,t}(x,y) = \max\{b_{\theta,t}(x,y), \epsilon\}$, and define inverse-surprisal weights

$$\widetilde{v}_{\theta,t}(x,y) = \frac{\tilde{b}_{\theta,t}(x,y)^{-1}}{\sum_{s=1}^{T_y} \tilde{b}_{\theta,s}(x,y)^{-1}}.$$

As with the high-surprisal token-weighted term, we use a per-example scale correction so that the z-free base model reconstruction remains on the same base-relative scale:

$$v_{\theta,t}(x,y) = \frac{\widetilde{v}_{\theta,t}(x,y)}{\eta_\theta(x,y)}, \qquad \eta_\theta(x,y) = \frac{\sum_{s=1}^{T_y} \widetilde{v}_{\theta,s}(x,y) b_{\theta,s}(x,y)}{b_\theta(x,y)}.$$

Replacing $a_{\theta,t}$ by $v_{\theta,t}$ emphasizes tokens already well predicted by p_θ, while preserving the same z-free reconstruction scale in expectation. The ablation therefore tests the mechanism-level hypothesis that alignment gains come from pressure on high-surprisal, strategy-relevant response regions rather than from nonuniform token weighting alone.

E.3 Training settings and hyperparameter sweeps

The default experimental setting is multi-task factorization: each run trains one router-generator model jointly across all six benchmark task families. This setting better matches the heterogeneous training regime of language models, where a single model must represent many task distributions. Additional single-task runs are used for diagnostic sweeps and mechanism plots where per-task variation is useful.

For the pretrained experiments, we initialize base models from Qwen2.5-0.5B and Qwen2.5-1.5B (Qwen Team, 2024), SmolLM2-360M (Allal et al., 2025), and OLMo 2 1B (OLMo et al., 2025). We also train base models from random initialization. For each, we first fit the model to the benchmark task distribution. We then freeze and factorize the resulting base model p_θ. The pretrained method comparisons use the multi-task setting and a continuous latent dimension of 64. The pretrained Hugging Face checkpoints are `Qwen/Qwen2.5-0.5B`, `Qwen/Qwen2.5-1.5B`, `HuggingFaceTB/SmolLM2-360M`, and `allenai/OLMo-2-0425-1B`; all four model cards list the Apache 2.0 license. The randomly initialized diagnostic grid contains 504 complete LoRA adaptation runs: six single-task settings plus the multi-task setting, twelve reconstruction-family settings including ablations, three KL weights $\beta \in \{1, 0.1, 0.01\}$, and two schedules for β (constant and linear warmup over training). For the Qwen2.5 comparisons, we select the best run for each objective from sweeps over the KL-to-reconstruction weight β. For SmolLM2-360M and OLMo 2 1B, we evaluate the six fixed objective settings selected for the paper, without repeating the sweeps. Each pretrained model and objective has one selected training run. Task-level points in the figures describe variation across benchmark tasks.

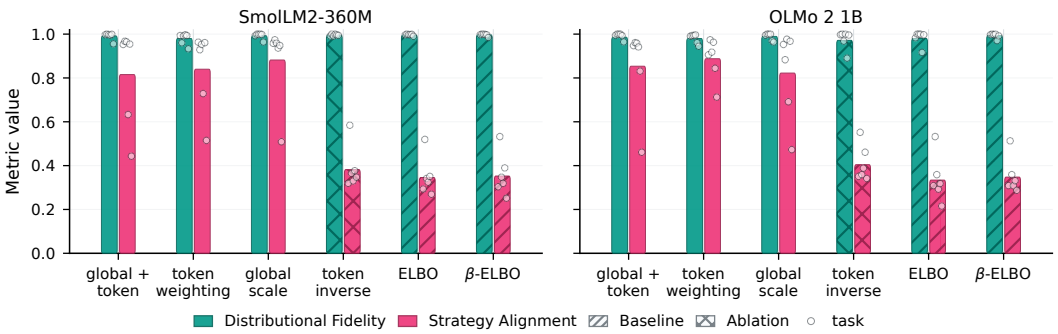


Figure 10: **Method comparison for SmolLM2-360M and OLMo 2 1B.** Distributional Fidelity and cross-input Strategy Alignment across different objectives.



Figure 11: **Task-faceted posterior latent dynamics.** Each row shows one benchmark task for the selected multi-task token-weighting run, and columns show checkpoints over training. Points are posterior mean latents projected with task-local PCA and colored by reference strategy. The view complements Figure 7 by showing that strategy organization emerges within each task family.

E.4 Pretrained base-model method comparison

E.5 Additional diagnostics

The main text focuses on the headline pretrained method comparison and compact mechanism diagnostics. Here we include additional views that support the same claims: latent geometry over training, role of KL weights, metric associations, and phase-space training dynamics.

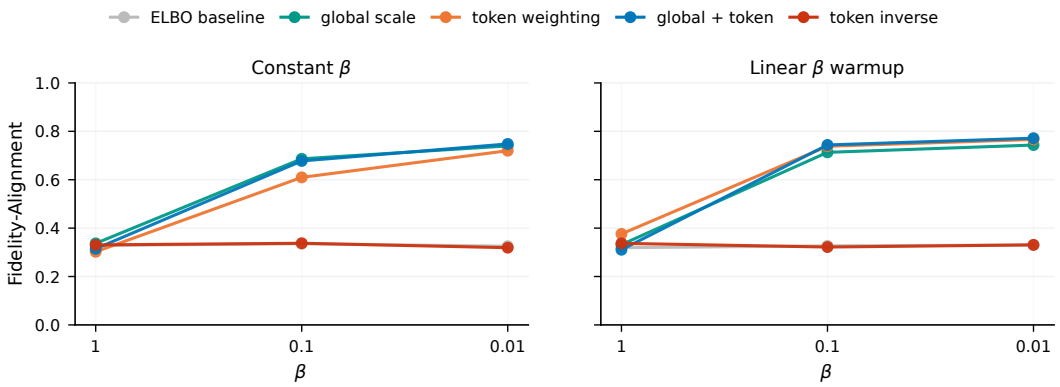


Figure 12: **Method success across β settings.** Plots show the product of Distributional Fidelity and Strategy Alignment across different KL weights β for the multi-task randomly initialized diagnostic grid. Standard ELBO-style baselines remain weak in fidelity-alignment across β, while model-directed objectives succeed robustly once β is small enough to enable informative latents.

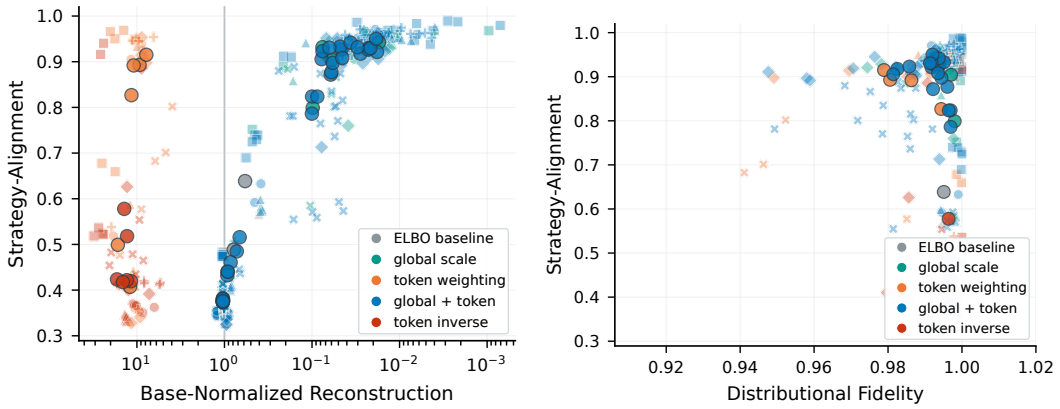


Figure 13: **Endpoint metric associations in single-task diagnostic runs. Left:** base-relative reconstruction versus Strategy Alignment. There is a general positive association between reconstruction measured in base model units and strategy alignment; however, unlike the token-weighted reconstruction shown in the main text, the association is weaker for the pure token-weighted variant. This is likely caused by too little reconstruction pressure on non high-surprisal tokens, which causes some loss on these tokens. This motivates setting $\alpha > 0$ in the global+token variant to preserve some uniform reconstruction pressure. **Right:** Distributional Fidelity versus Strategy Alignment. Larger outlined points show method centroids and smaller translucent points show individual runs; these diagnostics illustrate that it is possible to have high distributional fidelity without strategy alignment. In general distributional fidelity is easy, especially because the factorization is initialized near the pretrained model's response distribution, but strategy alignment is harder and requires the right objective design.

E.6 Exploration protocol and supporting results

Algorithm 2 describes our procedure for actively exploring latent space to enable exploration in strategy space under the model's generations. The algorithm biases router generations towards underexplored regions of latent space (i.e., towards latents that are farther away from previously explored latents). This is done under a whitened gaussian distribution. Notably, this procedure depends only on previously selected latents, not on decoded responses.

We compare this method to two baselines: (1) IID generation from the (unfactorized) base model; and (2) IID generation from the router-generator factorization, without active exploration in latent space (i.e., each generation samples the latent independently of previously sampled latents). All methods sample responses with temperature 1.0 and top-k 50. We report task-level cumulative

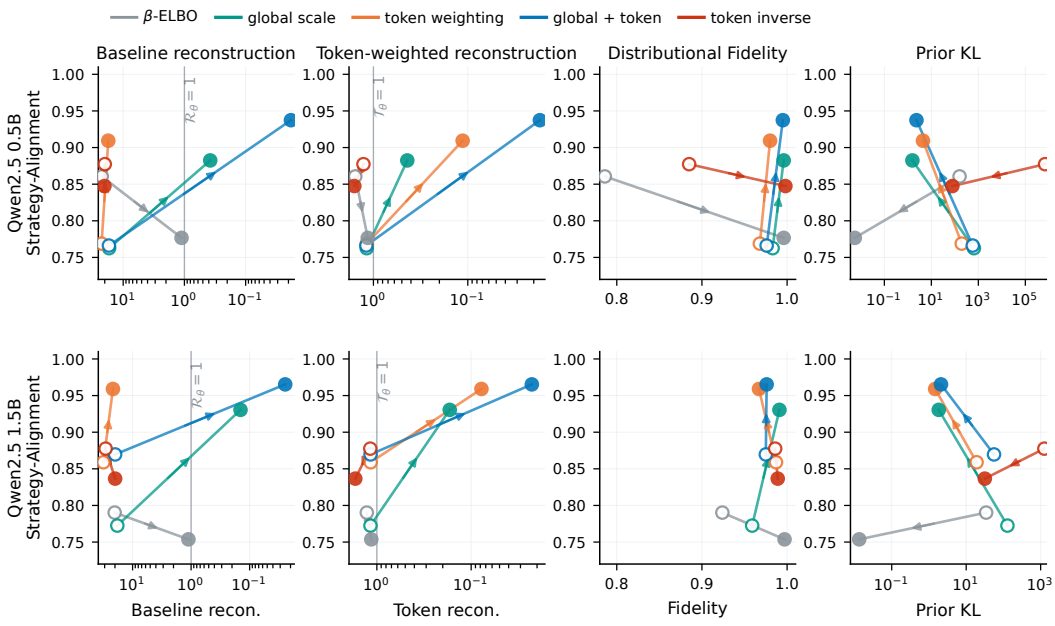


Figure 14: **Phase-space training dynamics of pretrained Qwen model factorization.** Phase-space dynamics diagrams of Qwen2.5 0.5B and 1.5B multi-task runs are shown across base-relative reconstruction, token-weighted reconstruction, Distributional Fidelity, and Prior KL, each plotted against Strategy Alignment. Trajectories show how each method moves through these diagnostic phase spaces over training, from initialization to the final checkpoint.

Algorithm 2: Latent Space Active Exploration

Input: Input x; router $r_\phi(z|x) = \mathcal{N}(\mu_x, \mathrm{diag}(\sigma_x^2))$; generator g_ϕ; response budget T; candidates per step K.
Output: Responses $y_1, \ldots, y_T$.

Define $d(u, v) = 1 - u^\top v / (\|u\|_2 \|v\|_2)$
for $t = 1, \ldots, T$ **do**
  Draw $U_t = \{u_{t,1}, \ldots, u_{t,K}\}$, with $u_{t,k} \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$
  if $t = 1$ **then**
    Set $u_t = u_{t,1}$
  else
    Set $u_t = \arg\max_{u \in U_t} \min_{s<t} d(u, u_s)$
  end
  Set $z_t = \mu_x + \sigma_x \odot u_t$
end
for $t = 1, \ldots, T$ **do**
  Sample $y_t \sim g_\phi(\cdot|x, z_t)$
end
return $y_1, \ldots, y_T$

strategy coverage: the number of distinct strategies found divided by the number available for that task, as a function of the sampling budget T.

Results are shown in Figure 15. IID sampling from the base model and router-generator factorizations (without latent space exploration) yield similar strategy coverage, as expected. For the inverse-surprisal factorization ablation, the coverage is *lower* than the base model. At a sampling budget of $T = 4$, the active latent exploration method yields a gain of about 7–8 percentage points for for the model-directed factorizations, while the ELBO controls and token-inverse ablation have no gain. The advantage is strongest at small sample budgets and narrows as coverage approaches saturation.

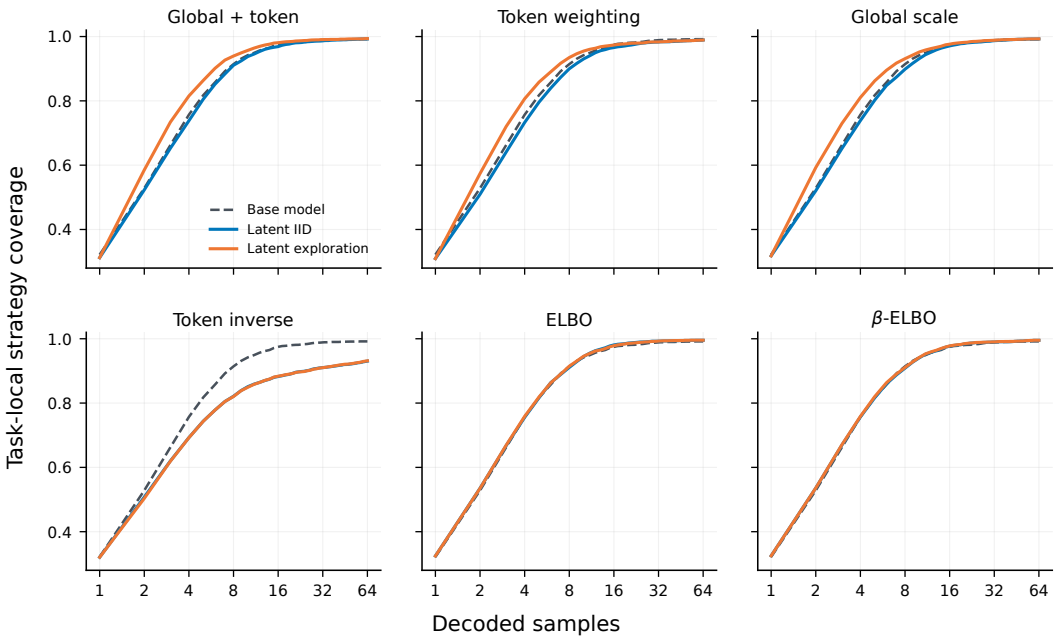


Figure 15: **Strategy coverage across decoding budgets.** Stochastic base-model sampling, Latent IID, and whitened angular selection are compared for all six from-scratch objective variants under the same temperature 1.0, top-k 50 decoding policy. The horizontal axis is logarithmic in decoded samples to make early-budget differences visible; all methods are evaluated through 64 samples.

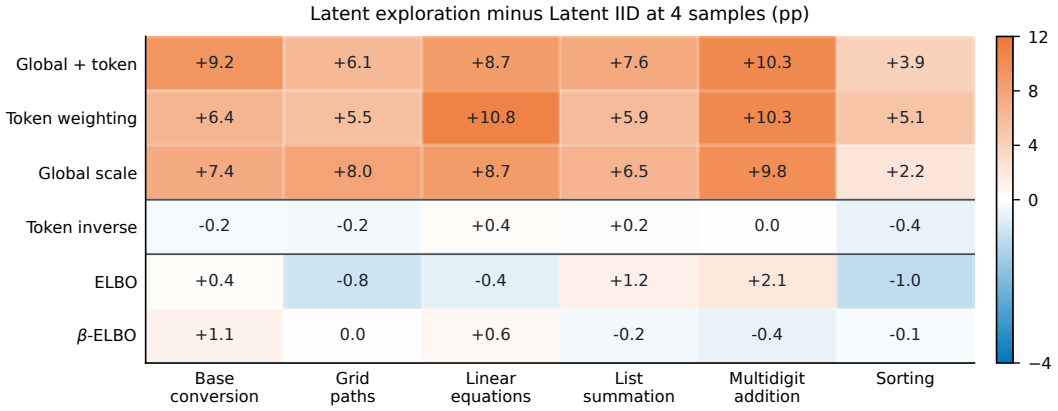


Figure 16: **Task-level benefit of active latent selection.** Each cell is the paired difference in task-local strategy coverage after four stochastic decoded samples, Latent Exploration minus Latent IID, in percentage points. Columns are the six benchmark tasks; rows are the same from-scratch objective variants as in the main-text figure. Task sample counts range from 157 to 187 inputs. Values are descriptive point estimates from one checkpoint per objective.

Figure 16 breaks the four-sample gain down by task. The three model-directed objectives have positive paired point estimates on every task, ranging from 2.2 to 10.8 percentage points.

F A Theory of Variational Approaches to the Strategy-Structure Recovery Problem

This appendix gives the formal treatment behind the preview theorem in the main text. The main result studies the population ELBO over independently parameterized expressive encoder, router, and generator classes.

F.1 Setup and the Strategy Structure Recovery Problem

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets (e.g., space of token sequences). Let μ be a fixed input law on $\mathcal{X}$, and let $p(y \mid x)$ be a fixed conditional law on $\mathcal{Y}$. All expectations, entropies, and mutual informations are computed under

$$X \sim \mu, \qquad Y \sim p(\cdot \mid X),$$

unless stated otherwise. Because all sample spaces are finite, measurability is automatic; the only analytic issues are support, finite-value, and minimizer-attainment issues, stated explicitly below.

Fix a latent cardinality K, and write $\mathcal{Z} = [K]$. Let $\mathcal{Q}, \mathcal{R}, \mathcal{G}$ be classes of encoders $q(z \mid x, y)$, routers $r(z \mid x)$, and generators $g(y \mid x, z)$, respectively. For $q \in \mathcal{Q}$, define the joint law

$$Q_q(x, y, z) = \mu(x)p(y \mid x)q(z \mid x, y).$$

Under Q_q, define the encoder-induced router

$$q(z \mid x) := \sum_y p(y \mid x)q(z \mid x, y),$$

and the encoder-induced branch conditional, whenever $q(z \mid x) > 0$, by

$$p_q(y \mid x, z) := \frac{p(y \mid x)q(z \mid x, y)}{q(z \mid x)}.$$

The conditional $p_q(y \mid x, z)$ may be defined arbitrarily on zero-mass branches; all KL and expectation terms below weight such branches by $q(z \mid x)$. Let $I_q(Y; Z \mid X)$ denote conditional mutual information under Q_q.

Let $S^\star = \sigma(X, Y) \in [K]$ denote the reference strategy. This hard-strategy assumption says that, once the completed trace is observed, the strategy is recognizable as an attribute of (X, Y). Let $q^\star$ be the hard reference encoder

$$q^\star(z \mid x, y) := \mathbf{1}\{z = \sigma(x, y)\}.$$

We assume $q^\star \in \mathcal{Q}$ when stating recovery theorems.

Because latent labels are arbitrary, recovery can only be defined up to relabeling. The next definition separates relabelings that may vary with the input from a single label system shared across inputs.

Definition 1: Weak and strong recovery

An encoder $q \in \mathcal{Q}$ weakly recovers $S^\star$ if, for μ-almost every x, there exists a permutation π_x of $[K]$ such that

$$Z = \pi_x(S^\star) \qquad Q_q\text{-almost surely conditional on } X = x.$$

It strongly recovers $S^\star$ if there exists one global permutation π of $[K]$ such that $Z = \pi(S^\star)$ Q_q-almost surely. Null labels are ignored; any bijection between positive-probability labels may be extended arbitrarily to a permutation of $[K]$.

We measure approximate recovery by the corresponding variation-of-information distances under Q_q:

$$d_{\mathrm{w}}(q, S^\star) := H_q(Z \mid X, S^\star) + H_q(S^\star \mid X, Z),$$

and

$$d_{\mathrm{s}}(q, S^\star) := H_q(Z \mid S^\star) + H_q(S^\star \mid Z).$$

The weak distance conditions on X, so it compares strategy semantics separately within each input. The strong distance does not condition on X, so it also penalizes input-dependent relabeling. The following lemma verifies that these distances are exactly the right zero-error notions for the two definitions.

Lemma 1: Recovery distances

For every $q \in \mathcal{Q}$:

1. $d_{\mathrm{w}}(q, S^\star) = 0$ if and only if q weakly recovers $S^\star$.
2. $d_{\mathrm{s}}(q, S^\star) = 0$ if and only if q strongly recovers $S^\star$.
3. $d_{\mathrm{w}}(q, S^\star) \le d_{\mathrm{s}}(q, S^\star)$.

Proof. For the weak statement, $d_{\mathrm{w}}(q, S^\star) = 0$ holds if and only if

$$H_q(Z \mid X, S^\star) = 0, \qquad H_q(S^\star \mid X, Z) = 0.$$

Thus, for μ-almost every x, Z is a deterministic function of $S^\star$ under $Q_q(\cdot \mid X = x)$, and $S^\star$ is a deterministic function of Z under the same conditional law. The two variables therefore induce the same partition of positive-probability labels at that input, up to a bijection between the labels that occur. Since both label spaces are subsets of $[K]$, the bijection extends to a permutation π_x. This is weak recovery. The converse is immediate because $Z = \pi_x(S^\star)$ makes the two conditional entropies vanish.

The strong statement is the same argument without conditioning on X. The condition $d_{\mathrm{s}}(q, S^\star) = 0$ is equivalent to Z and $S^\star$ determining each other as global random variables under Q_q, which is strong recovery up to null labels.

Finally, conditioning reduces entropy:

$$H_q(Z \mid X, S^\star) \le H_q(Z \mid S^\star), \qquad H_q(S^\star \mid X, Z) \le H_q(S^\star \mid Z).$$

Adding these inequalities gives $d_{\mathrm{w}}(q, S^\star) \le d_{\mathrm{s}}(q, S^\star)$. □

Thus any strong recovery guarantee also implies weak recovery, but not conversely. This distinction becomes important when different inputs can reuse the same latent labels with different meanings.

F.2 ELBO Decomposition with Router-Generator Complexity

We study the population complexity-penalized beta-ELBO, with $\beta \ge 0$ and $\lambda \ge 0$,

$$\mathcal{L}_{\beta,\lambda}(q, r, g) := \mathbb{E}_{Q_q}[-\log g(Y \mid X, Z)] + \beta\, \mathbb{E}_{X,Y}\mathrm{KL}(q(\cdot \mid X, Y) \| r(\cdot \mid X)) + \lambda C(r, g).$$

The functional $C(r, g) \in [0, \infty]$ is a theoretical proxy for architecture, parameterization, and optimization biases that favor simple router-generator factorizations. We assume it is invariant under global relabeling of latent states. All KL terms use the usual extended-real convention.

Define

$$\mathrm{DecGap}(q, g) := \mathbb{E}_{X,Z}\mathrm{KL}(p_q(\cdot \mid X, Z) \| g(\cdot \mid X, Z)),$$

and

$$\mathrm{PriorGap}(q, r) := \mathbb{E}_X \mathrm{KL}(q(\cdot \mid X) \| r(\cdot \mid X)).$$

The following lemma decomposes the variational objective into four terms: the irreducible conditional entropy $H(Y \mid X)$, an information term involving $I_q(Y; Z \mid X)$, the router and decoder KL gaps, and the complexity penalty $C(r, g)$. This form makes clear how the objective trades off information in the latent variable against the fit and complexity of the router-generator factorization.

Lemma 2: ELBO information-complexity decomposition

For every (q, r, g),

$$\mathcal{L}_{\beta,\lambda}(q, r, g) = H(Y \mid X) + (\beta - 1) I_q(Y; Z \mid X) + \mathrm{DecGap}(q, g) + \beta\, \mathrm{PriorGap}(q, r) + \lambda C(r, g).$$

Proof. First decompose the reconstruction term. Under Q_q,

$$\mathbb{E}[-\log g(Y \mid X, Z)] = \mathbb{E}_{X,Z} \sum_y p_q(y \mid X, Z)(-\log g(y \mid X, Z)).$$

For each (x, z), cross-entropy equals entropy plus KL:

$$\sum_y p_q(y \mid x, z)(-\log g(y \mid x, z)) = H(p_q(\cdot \mid x, z)) + \mathrm{KL}(p_q(\cdot \mid x, z) \| g(\cdot \mid x, z)).$$

Averaging gives

$$\mathbb{E}[-\log g(Y \mid X, Z)] = H_q(Y \mid X, Z) + \mathrm{DecGap}(q, g).$$

Because the (X, Y) marginal of Q_q is $\mu(x)p(y \mid x)$,

$$H_q(Y \mid X) = H(Y \mid X).$$

Thus

$$H_q(Y \mid X, Z) = H(Y \mid X) - I_q(Y; Z \mid X),$$

and

$$\mathbb{E}[-\log g(Y \mid X, Z)] = H(Y \mid X) - I_q(Y; Z \mid X) + \mathrm{DecGap}(q, g).$$

Next decompose the KL-to-prior term:

$$\mathbb{E}_{X,Y} \mathrm{KL}(q(\cdot \mid X, Y) \| r(\cdot \mid X)) = \mathbb{E}_{Q_q} \log \frac{q(Z \mid X, Y)}{r(Z \mid X)}.$$

Insert $q(Z \mid X)$:

$$\log \frac{q(Z \mid X, Y)}{r(Z \mid X)} = \log \frac{q(Z \mid X, Y)}{q(Z \mid X)} + \log \frac{q(Z \mid X)}{r(Z \mid X)}.$$

The first expectation is $I_q(Y; Z \mid X)$, and the second is $\mathrm{PriorGap}(q, r)$. Therefore

$$\mathbb{E}_{X,Y} \mathrm{KL}(q(\cdot \mid X, Y) \| r(\cdot \mid X)) = I_q(Y; Z \mid X) + \mathrm{PriorGap}(q, r).$$

Combining the reconstruction decomposition, the KL decomposition, and the complexity term gives the identity. □

The coefficient of $I_q(Y; Z \mid X)$ is $\beta - 1$. Therefore the objective is information-seeking only when $\beta < 1$. At $\beta = 1$, selection can still occur through the profiled cost below, but the ELBO no longer rewards conditional information after the variational gaps have been profiled out. For $\beta > 1$, the objective penalizes information in Z.

F.3 Profiled Router-Generator Objective

For a fixed encoder q, define the profiled router-generator cost

$$A_{\beta,\lambda}(q) := \inf_{r \in \mathcal{R},\ g \in \mathcal{G}} [\mathrm{DecGap}(q, g) + \beta\, \mathrm{PriorGap}(q, r) + \lambda C(r, g)] \in [0, \infty].$$

Also define the profiled population loss

$$\overline{\mathcal{L}}_{\beta,\lambda}(q) := \inf_{r \in \mathcal{R},\ g \in \mathcal{G}} \mathcal{L}_{\beta,\lambda}(q, r, g).$$

Profiling asks how good an encoder can look after choosing the best available router and generator for it. The next proposition rewrites that profiled loss as an encoder-only score.

Proposition 1: Profiled ELBO score

For every $q \in \mathcal{Q}$,

$$\overline{\mathcal{L}}_{\beta,\lambda}(q) = H(Y \mid X) - (1-\beta) I_q(Y; Z \mid X) + A_{\beta,\lambda}(q).$$

If $\beta < 1$, define

$$J^{\mathrm{prof}}_{\beta,\lambda}(q) := I_q(Y; Z \mid X) - \frac{A_{\beta,\lambda}(q)}{1-\beta},$$

Then

$$\overline{\mathcal{L}}_{\beta,\lambda}(q) = H(Y \mid X) - (1-\beta) J^{\mathrm{prof}}_{\beta,\lambda}(q).$$

Proof. Take the infimum over $r \in \mathcal{R}$ and $g \in \mathcal{G}$ in Lemma 2. The terms $H(Y \mid X)$ and $I_q(Y; Z \mid X)$ do not depend on r or g, and the remaining infimum is exactly $A_{\beta,\lambda}(q)$. The final identity is a rearrangement when $\beta < 1$. □

Thus the full expressive ELBO selects encoders by an information-minus-profiled approximation-complexity score. The quantity $A_{\beta,\lambda}(q)$, which involves the router-generator complexity $C(r, g)$, measures whether the encoder-induced routing law and branch conditionals can be approximated with low profiled cost by a simple router-generator pair.

F.4 Metric Recovery Margins

Fix a notion of recovery $\rho \in \{\mathrm{w}, \mathrm{s}\}$ and $\delta > 0$. Define the bad encoder set

$$\mathcal{B}_\rho(\delta) := \{q \in \mathcal{Q} : d_\rho(q, S^\star) \geq \delta\}.$$

For $\beta < 1$ and $A_{\beta,\lambda}(q^\star) < \infty$, define the profiled metric recovery margin

$$\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta) := J^{\mathrm{prof}}_{\beta,\lambda}(q^\star) - \sup_{q \in \mathcal{B}_\rho(\delta)} J^{\mathrm{prof}}_{\beta,\lambda}(q),$$

with the convention that the supremum of an empty set is $-\infty$. Equivalently,

$$\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta) = \inf_{q \in \mathcal{B}_\rho(\delta)} \left[I^\star - I(q) + \frac{A_{\beta,\lambda}(q) - A_{\beta,\lambda}(q^\star)}{1-\beta} \right],$$

where $I(q) := I_q(Y; Z \mid X)$ and $I^\star := I_{q^\star}(Y; Z \mid X)$.

The following theorem states the population strategy-structure recovery criterion. When the profiled margin at level δ is positive, every population ELBO minimizer must recover the reference strategy up to recovery distance δ.

Theorem 1: Population strategy-structure recovery for the expressive ELBO

Assume $\beta < 1$, $q^\star \in \mathcal{Q}$, $A_{\beta,\lambda}(q^\star) < \infty$, and $\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta) > 0$. If a profiled minimizer exists and

$$\widehat{q} \in \arg\min_{q \in \mathcal{Q}} \overline{\mathcal{L}}_{\beta,\lambda}(q),$$

then

$$d_\rho(\widehat{q}, S^\star) < \delta.$$

Consequently, if a full ELBO minimizer exists and

$$(\widehat{q}, \widehat{r}, \widehat{g}) \in \arg\min_{q \in \mathcal{Q},\, r \in \mathcal{R},\, g \in \mathcal{G}} \mathcal{L}_{\beta,\lambda}(q, r, g),$$

then $d_\rho(\widehat{q}, S^\star) < \delta$.

Proof. By Proposition 1, minimizing $\overline{\mathcal{L}}_{\beta,\lambda}$ is equivalent to maximizing $J^{\mathrm{prof}}_{\beta,\lambda}$ when $\beta < 1$. Hence any profiled population minimizer $\widehat{q}$ satisfies

$$J^{\mathrm{prof}}_{\beta,\lambda}(\widehat{q}) \geq J^{\mathrm{prof}}_{\beta,\lambda}(q^\star).$$

If $d_\rho(\widehat{q}, S^\star) \geq \delta$, then $\widehat{q} \in \mathcal{B}_\rho(\delta)$, so

$$J^{\mathrm{prof}}_{\beta,\lambda}(\widehat{q}) \leq \sup_{q \in \mathcal{B}_\rho(\delta)} J^{\mathrm{prof}}_{\beta,\lambda}(q) = J^{\mathrm{prof}}_{\beta,\lambda}(q^\star) - \Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta) < J^{\mathrm{prof}}_{\beta,\lambda}(q^\star),$$

contradicting optimality. Therefore $d_\rho(\widehat{q}, S^\star) < \delta$.

For the final claim, let $(\widehat{q}, \widehat{r}, \widehat{g})$ be a full ELBO minimizer and let

$$V := \inf_{q \in \mathcal{Q},\, r \in \mathcal{R},\, g \in \mathcal{G}} \mathcal{L}_{\beta,\lambda}(q, r, g) = \mathcal{L}_{\beta,\lambda}(\widehat{q}, \widehat{r}, \widehat{g}).$$

For every q, $V \leq \overline{\mathcal{L}}_{\beta,\lambda}(q)$, so $V \leq \inf_q \overline{\mathcal{L}}_{\beta,\lambda}(q)$. Conversely,

$$\inf_q \overline{\mathcal{L}}_{\beta,\lambda}(q) \leq \overline{\mathcal{L}}_{\beta,\lambda}(\widehat{q}) \leq \mathcal{L}_{\beta,\lambda}(\widehat{q}, \widehat{r}, \widehat{g}) = V.$$

Thus $\widehat{q}$ minimizes the profiled loss over $\mathcal{Q}$, and the first part applies. □

This theorem reduces population recovery to a score-separation condition. It states that semantic recovery of the reference strategy is controlled by the profiled-score gap $\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)$: if this gap is positive, then optimizing the complexity-penalized variational objective selects only encoders inside the desired δ-recovery neighborhood.

Remark 1: Sufficient conditions for minimizer existence

The above theorem is conditional on minimizers existing. One sufficient route, in these finite ambient spaces, is that $\mathcal{Q}, \mathcal{R}, \mathcal{G}$ are compact subsets of the relevant probability simplexes and C is lower semicontinuous, with support restrictions or closure conventions that make the KL terms lower semicontinuous and avoid undefined $\infty - \infty$ expressions. The substantive content of this theorem is the identification of the profiled score that must separate the target encoder from misaligned encoders.

The margin definition is sometimes hard to check directly because it ranges over all bad encoders. The following corollary gives a more interpretable sufficient condition in terms of an information advantage and a profiled-cost advantage. It says that if encoders that fail to recover the strategy don't have much more conditional information than the target, and they pay a large enough profiled cost penalty, then the selected encoder will recover the strategy.

Corollary 1: A sufficient information-complexity separation condition

Assume the hypotheses of Theorem 1 except for the positivity of the margin. Suppose there exist $\gamma_I \in \mathbb{R}$ and $\gamma_A > 0$ such that, for every $q \in \mathcal{B}_\rho(\delta)$,

$$I_q(Y; Z \mid X) - I_{q^\star}(Y; Z \mid X) \leq \gamma_I,$$

and

$$A_{\beta,\lambda}(q) - A_{\beta,\lambda}(q^\star) \geq \gamma_A.$$

If $\gamma_A > (1-\beta)\gamma_I$, then

$$\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta) \geq \frac{\gamma_A}{1-\beta} - \gamma_I > 0.$$

Proof. For every bad q,

$$J^{\mathrm{prof}}_{\beta,\lambda}(q^\star) - J^{\mathrm{prof}}_{\beta,\lambda}(q) = I_{q^\star}(Y; Z \mid X) - I_q(Y; Z \mid X) + \frac{A_{\beta,\lambda}(q) - A_{\beta,\lambda}(q^\star)}{1-\beta} \geq -\gamma_I + \frac{\gamma_A}{1-\beta}.$$

Taking the infimum over $\mathcal{B}_\rho(\delta)$ proves the claim. □

This corollary makes the tradeoff explicit: a bad encoder may have more conditional information, but only up to γ_I. The target still wins if the bad encoder pays a large enough additional router-generator cost.

We now return to the distinction between weak and strong recovery. Strong recovery requires more than recovering the correct partition separately for each input; it also requires label alignment *across* inputs. The next proposition shows that input-dependent relabelings can preserve weak recovery and conditional information exactly, while still preventing strong recovery unless the profiled cost favors a globally consistent label system.

Proposition 2: Input-dependent relabelings and strong recovery

For a family of permutations $\pi = \{\pi_x : x \in \mathcal{X}\}$, define the hard encoder

$$q^\pi(z \mid x, y) := \mathbf{1}\{z = \pi_x(\sigma(x, y))\}.$$

Assume $\beta < 1$. If $q^\pi \in \mathcal{Q}$, then $d_{\mathrm{w}}(q^\pi, S^\star) = 0$ and

$$I_{q^\pi}(Y; Z \mid X) = I_{q^\star}(Y; Z \mid X).$$

If no single global permutation agrees with π_x on the positive-probability support of $S^\star \mid X = x$ for μ-almost every x, then $d_{\mathrm{s}}(q^\pi, S^\star) > 0$. In that case, $q^\star$ beats this input-dependent relabeling in the profiled score if and only if

$$A_{\beta,\lambda}(q^\pi) > A_{\beta,\lambda}(q^\star).$$

Consequently, strong recovery requires a positive profiled-cost gap against such non-globally aligned relabelings; if some such q^π has $A_{\beta,\lambda}(q^\pi) = A_{\beta,\lambda}(q^\star)$, then the strong margin cannot be positive for every $\delta \le d_{\mathrm{s}}(q^\pi, S^\star)$.

Proof. Conditional on $X = x$, the map $S^\star \mapsto \pi_x(S^\star)$ is a bijection on labels, so weak recovery holds and

$$I_{q^\pi}(Y; Z \mid X) = H_{q^\pi}(Z \mid X) = H(S^\star \mid X) = I_{q^\star}(Y; Z \mid X).$$

If no global permutation agrees with the family $\{\pi_x\}$ on the positive supports, then q^π does not strongly recover $S^\star$; by Lemma 1, $d_{\mathrm{s}}(q^\pi, S^\star) > 0$. Since the conditional information terms are equal, the profiled-score comparison between $q^\star$ and q^π is determined exactly by the profiled costs. If the profiled costs tie, the two scores tie, so no positive strong margin can separate $q^\star$ from q^π at any scale including q^π in the bad set. □

This shows why strong recovery requires more than predictive information. The model class or complexity proxy must also prefer a globally aligned label system over equally informative local relabelings.

The recovery theorem above gives approximate recovery at a chosen radius with respect to the corresponding recovery distance. When the candidate class has a positive distance gap around the target equivalence class, that approximate statement upgrades to exact recovery.

Corollary 2: Exact recovery from metric recovery

Let $\sim_\rho$ denote weak equivalence when $\rho = \mathrm{w}$ and strong equivalence when $\rho = \mathrm{s}$. Suppose

$$m_\rho := \inf_{q \not\sim_\rho q^\star} d_\rho(q, S^\star) > 0.$$

If Theorem 1 holds for some $\delta \le m_\rho$, then every population ELBO minimizer exactly recovers $S^\star$ in the corresponding weak or strong sense.

Proof. The theorem gives $d_\rho(\widehat{q}, S^\star) < \delta \le m_\rho$. By definition of m_ρ, every non-recovering encoder has distance at least m_ρ. Therefore $\widehat{q} \sim_\rho q^\star$. □

The separation assumption $m_\rho > 0$ is natural for finite or otherwise discrete encoder classes.

Example 1: Positive margin from semantic branch sharing

Suppose the target strategy yields branch conditionals that can be represented by shared low-complexity generator templates, while every encoder at recovery scale δ either has at most γ_I more conditional information or requires at least γ_A additional profiled router-generator cost. If $\gamma_A > (1-\beta)\gamma_I$, Corollary 1 gives a positive recovery margin. This is the intended regime: semantic strategies win not merely because they are informative, but because their factorization is cheaper to realize.

Example 2: Failure from a superficial high-information attribute

If $\lambda = 0$ and $\mathcal{R}, \mathcal{G}$ are expressive enough that $A_{\beta,\lambda}(q) = 0$ for all candidate deterministic encoders, then the profiled score is just $I_q(Y; Z \mid X)$. A length bucket, formatting marker, or balanced nuisance partition with larger conditional information than $S^\star$ will beat the semantic strategy. The theorem then correctly predicts non-recovery.

Example 3: Failure of strong recovery under free local relabeling

If the available router-generator class and complexity proxy are invariant not only to global relabeling but also to input-dependent relabeling, then the encoders q^π in Proposition 2 can tie $q^\star$ in profiled score. Weak recovery may still hold, but the strong margin cannot be positive unless the objective assigns positive cost to such local label misalignment.

F.5 Marginal Fidelity and the Small-Complexity Limit

For a router-generator pair, define the marginal conditional law

$$m_{r,g}(y \mid x) := \sum_z r(z \mid x) g(y \mid x, z).$$

The next proposition connects the profiled variational gaps to the observable marginal fit of this router-generator pair. It clarifies how much fidelity can be lost when the profiled optimizer trades exact factorization for lower complexity.

Proposition 3: Variational gaps control marginal fidelity

For every (q, r, g),

$$\mathbb{E}_X \mathrm{KL}(p(\cdot \mid X) \| m_{r,g}(\cdot \mid X)) \le \mathrm{PriorGap}(q, r) + \mathrm{DecGap}(q, g).$$

Proof. Fix $x \in \mathcal{X}$. Define two joint distributions on $\mathcal{Y} \times \mathcal{Z}$:

$$P_x(y, z) := p(y \mid x) q(z \mid x, y), \qquad M_x(y, z) := r(z \mid x) g(y \mid x, z).$$

The distribution P_x is the joint law over (Y, Z) induced by first sampling $Y \sim p(\cdot \mid x)$ and then $Z \sim q(\cdot \mid x, Y)$. By Bayes' rule, it can also be written as

$$P_x(y, z) = q(z \mid x) p_q(y \mid x, z),$$

with the usual convention that zero-mass branches do not affect the KL terms.

Now compare the Y-marginals of these two joint distributions. For P_x,

$$\sum_z P_x(y, z) = \sum_z p(y \mid x) q(z \mid x, y) = p(y \mid x).$$

For M_x,

$$\sum_z M_x(y, z) = \sum_z r(z \mid x) g(y \mid x, z) = m_{r,g}(y \mid x).$$

Thus $p(\cdot \mid x)$ and $m_{r,g}(\cdot \mid x)$ are obtained from P_x and M_x, respectively, by applying the same projection map $(y, z) \mapsto y$.

By data processing for KL divergence, marginalizing out Z cannot increase KL. Therefore

$$\mathrm{KL}(p(\cdot \mid x) \| m_{r,g}(\cdot \mid x)) \le \mathrm{KL}(P_x \| M_x).$$

It remains to expand the joint KL on the right. Using the factorizations $P_x(y, z) = q(z \mid x)p_q(y \mid x, z)$ and $M_x(y, z) = r(z \mid x)g(y \mid x, z)$,

$$\begin{aligned}
\mathrm{KL}(P_x \| M_x) &= \sum_{z,y} q(z \mid x)p_q(y \mid x, z) \log \frac{q(z \mid x)p_q(y \mid x, z)}{r(z \mid x)g(y \mid x, z)} \\
&= \sum_{z} q(z \mid x) \log \frac{q(z \mid x)}{r(z \mid x)} + \sum_{z} q(z \mid x) \sum_{y} p_q(y \mid x, z) \log \frac{p_q(y \mid x, z)}{g(y \mid x, z)} \\
&= \mathrm{KL}(q(\cdot \mid x) \| r(\cdot \mid x)) + \mathbb{E}_{Z \sim q(\cdot \mid x)} \mathrm{KL}(p_q(\cdot \mid x, Z) \| g(\cdot \mid x, Z)).
\end{aligned}$$

Combining the previous two displays gives, for each fixed x,

$$\mathrm{KL}(p(\cdot \mid x) \| m_{r,g}(\cdot \mid x)) \le \mathrm{KL}(q(\cdot \mid x) \| r(\cdot \mid x)) + \mathbb{E}_{Z \sim q(\cdot \mid x)} \mathrm{KL}(p_q(\cdot \mid x, Z) \| g(\cdot \mid x, Z)).$$

Finally, average both sides over $X \sim \mu$. The two terms on the right become $\mathrm{PriorGap}(q, r)$ and $\mathrm{DecGap}(q, g)$, respectively, which proves

$$\mathbb{E}_X \mathrm{KL}(p(\cdot \mid X) \| m_{r,g}(\cdot \mid X)) \le \mathrm{PriorGap}(q, r) + \mathrm{DecGap}(q, g).$$

□

This proposition makes the fidelity tradeoff explicit. For finite λ, the profiled optimizer can prefer a simpler (r, g) with nonzero variational gaps, and therefore need not exactly factorize $p(y \mid x)$. The next result records the high-fidelity limit.

Proposition 4: Small-λ fidelity for a fixed encoder

Fix $q \in \mathcal{Q}$ and $\beta > 0$. Suppose the exact induced pair

$$r_q(z \mid x) := q(z \mid x), \qquad g_q(y \mid x, z) := p_q(y \mid x, z)$$

belongs to $\mathcal{R} \times \mathcal{G}$, and $C(r_q, g_q) < \infty$. If (r_λ, g_λ) minimizes $A_{\beta,\lambda}(q)$, for each $\lambda > 0$, then

$$\mathrm{DecGap}(q, g_\lambda) \to 0, \qquad \mathrm{PriorGap}(q, r_\lambda) \to 0$$

as $\lambda \downarrow 0$. Consequently,

$$\mathbb{E}_X \mathrm{KL}(p(\cdot \mid X) \| m_{r_\lambda, g_\lambda}(\cdot \mid X)) \to 0.$$

Proof. Fix $q \in \mathcal{Q}$. By assumption, the exact induced router-generator pair

$$r_q(z \mid x) := q(z \mid x), \qquad g_q(y \mid x, z) := p_q(y \mid x, z)$$

belongs to $\mathcal{R} \times \mathcal{G}$ and has finite complexity.

For q's induced router-generator pair (r_q, g_q), the prior gap and decoder gap are both zero. Since (r_λ, g_λ) minimizes $A_{\beta,\lambda}(q)$, its objective value is no larger than the value achieved by the exact induced pair. Therefore

$$\begin{aligned}
&\mathrm{DecGap}(q, g_\lambda) + \beta\, \mathrm{PriorGap}(q, r_\lambda) + \lambda C(r_\lambda, g_\lambda) \\
&\quad \le \mathrm{DecGap}(q, g_q) + \beta\, \mathrm{PriorGap}(q, r_q) + \lambda C(r_q, g_q) = \lambda C(r_q, g_q).
\end{aligned}$$

All three terms on the left are nonnegative. Hence

$$\mathrm{DecGap}(q, g_\lambda) \le \lambda C(r_q, g_q),$$

and, since $\beta > 0$,

$$\mathrm{PriorGap}(q, r_\lambda) \le \frac{\lambda C(r_q, g_q)}{\beta}.$$

Because $C(r_q, g_q) < \infty$, both upper bounds go to zero as $\lambda \downarrow 0$. Thus

$$\mathrm{DecGap}(q, g_\lambda) \to 0, \qquad \mathrm{PriorGap}(q, r_\lambda) \to 0.$$

Finally, Proposition 3 gives

$$\mathbb{E}_X \mathrm{KL}(p(\cdot \mid X) \| m_{r_\lambda, g_\lambda}(\cdot \mid X)) \le \mathrm{PriorGap}(q, r_\lambda) + \mathrm{DecGap}(q, g_\lambda).$$

The right-hand side tends to zero, so the marginal KL also tends to zero. □

The limit $\lambda \downarrow 0$ restores high-fidelity factorization under exact realizability. It also weakens the selection role of complexity: the profiled encoder score approaches maximum conditional information, which can favor superficial high-information partitions unless complexity still breaks the relevant near-ties.

F.6 Empirical Recovery

The population theorem gives a profiled-score margin $\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)$. Empirical recovery only requires one additional ingredient: the empirical profiled ELBO must approximate the population profiled ELBO at a scale smaller than the same margin measured in loss units. Since

$$\overline{\mathcal{L}}_{\beta,\lambda}(q) = H(Y \mid X) - (1-\beta) J^{\mathrm{prof}}_{\beta,\lambda}(q),$$

the loss separation corresponding to $\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)$ is $(1-\beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)$. Equivalently, for every bad encoder $q \in \mathcal{B}_\rho(\delta)$,

$$\begin{aligned}\overline{\mathcal{L}}_{\beta,\lambda}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q^\star) = A_{\beta,\lambda}(q) - A_{\beta,\lambda}(q^\star) \\ - (1-\beta)\{I_q(Y;Z \mid X) - I_{q^\star}(Y;Z \mid X)\}.\end{aligned}$$

Thus empirical recovery is possible when finite samples preserve this information-complexity loss gap.

Let $(X_i, Y_i)_{i=1}^n$ be i.i.d. samples from $\mu(x)p(y \mid x)$. We use the analytic expectation over $z \sim q(\cdot \mid X_i, Y_i)$ and define

$$\begin{aligned}\widehat{\mathcal{L}}_{\beta,\lambda,n}(q, r, g) := \frac{1}{n}\sum_{i=1}^n \sum_z q(z \mid X_i, Y_i) \left[-\log g(Y_i \mid X_i, z) + \beta \log \frac{q(z \mid X_i, Y_i)}{r(z \mid X_i)} \right] \\ + \lambda C(r, g),\end{aligned}$$

with the usual extended-real conventions for the logarithmic terms. The empirical profiled loss is

$$\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) := \inf_{r \in \mathcal{R},\, g \in \mathcal{G}} \widehat{\mathcal{L}}_{\beta,\lambda,n}(q, r, g).$$

Theorem 2: Empirical recovery for the profiled ELBO

Assume $\beta < 1$, $q^\star \in \mathcal{Q}$, $\overline{\mathcal{L}}_{\beta,\lambda}(q^\star) < \infty$, and

$$\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta) > 0.$$

Suppose an empirical profiled minimizer exists,

$$\widehat{q}_n \in \arg\min_{q\in\mathcal{Q}} \widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q),$$

and the empirical profiled objective satisfies a uniform convergence bound:

$$\sup_{q\in\mathcal{Q}} \left|\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q)\right| < \frac{(1-\beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)}{2}.$$

Then every empirical profiled minimizer recovers $S^\star$ at scale δ:

$$d_\rho(\widehat{q}_n, S^\star) < \delta.$$

Proof. Let

$$\varepsilon_n := \sup_{q\in\mathcal{Q}} \left|\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q)\right|.$$

By empirical optimality,

$$\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(\widehat{q}_n) \le \widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q^\star).$$

Therefore

$$\begin{aligned}
\overline{\mathcal{L}}_{\beta,\lambda}(\widehat{q}_n) &\le \widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(\widehat{q}_n) + \varepsilon_n \\
&\le \widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q^\star) + \varepsilon_n \\
&\le \overline{\mathcal{L}}_{\beta,\lambda}(q^\star) + 2\varepsilon_n \\
&< \overline{\mathcal{L}}_{\beta,\lambda}(q^\star) + (1-\beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta).
\end{aligned}$$

If $d_\rho(\widehat{q}_n, S^\star) \ge \delta$, then $\widehat{q}_n \in \mathcal{B}_\rho(\delta)$. Since

$$\overline{\mathcal{L}}_{\beta,\lambda}(q) = H(Y \mid X) - (1-\beta)J^{\mathrm{prof}}_{\beta,\lambda}(q),$$

the definition of $\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)$ implies

$$\overline{\mathcal{L}}_{\beta,\lambda}(\widehat{q}_n) \ge \overline{\mathcal{L}}_{\beta,\lambda}(q^\star) + (1-\beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta),$$

a contradiction. Hence $d_\rho(\widehat{q}_n, S^\star) < \delta$. □

This theorem is deterministic conditional on uniform convergence. The remaining question is how large n must be for

$$\sup_{q\in\mathcal{Q}} \left|\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q)\right| < \frac{(1-\beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)}{2}$$

to hold with high probability. This is a statistical question about the profiled log-loss class.

In what follows, we give two concrete sufficient conditions for this uniform convergence. The first corollary assumes a finite encoder class and a fixed concentration bound for each encoder. The second derives a concrete condition under which such concentration holds by assuming a support floor for the decoder and router probabilities such that the log losses are bounded.

Corollary 3: Finite-class sample complexity

Suppose $|\mathcal{Q}| = N < \infty$ and for every fixed $q \in \mathcal{Q}$, the empirical profiled loss $\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q)$ satisfies

$$\mathbb{P}\left(\left|\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q)\right| \geq t\right) \leq 2\exp\left(-\frac{nt^2}{2\sigma^2}\right).$$

If

$$n \geq \frac{8\sigma^2}{(1-\beta)^2(\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta))^2}\log\frac{2N}{\eta},$$

then with probability at least $1-\eta$, every empirical profiled-ELBO minimizer satisfies $d_\rho(\widehat{q}_n, S^\star) < \delta$.

Proof. For each fixed $q \in \mathcal{Q}$, apply the assumed concentration inequality. A union bound over the N encoders gives

$$\mathbb{P}\left(\sup_{q\in\mathcal{Q}}\left|\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q)\right| \geq t\right) \leq 2N\exp\left(-\frac{nt^2}{2\sigma^2}\right).$$

Set

$$t = \frac{(1-\beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)}{2}.$$

The displayed lower bound on n makes the right-hand side at most η. On the complementary event,

$$\sup_{q\in\mathcal{Q}}\left|\widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q)\right| < \frac{(1-\beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)}{2}.$$

The empirical recovery theorem then implies $d_\rho(\widehat{q}_n, S^\star) < \delta$ for every empirical profiled-ELBO minimizer. □

Finite $\mathcal{Q}$ alone does not make the preceding concentration automatic. The log terms

$$-\log g(y \mid x, z), \qquad \log\frac{q(z \mid x, y)}{r(z \mid x)}$$

can be unbounded or infinite when $g(y \mid x, z)$ or $r(z \mid x)$ is near zero. A simple sufficient condition is to assume finite $\mathcal{Q}, \mathcal{R}, \mathcal{G}$, finite $C(r, g)$, and a support floor $\tau \in (0, 1]$ such that

$$r(z \mid x) \geq \tau, \qquad g(y \mid x, z) \geq \tau$$

for all admissible r, g and all x, y, z.

This is stated in the following corollary.

Corollary 4: Finite classes with bounded log loss

Suppose $\mathcal{Q}, \mathcal{R}, \mathcal{G}$ are finite, $C(r, g) < \infty$ for all $(r, g) \in \mathcal{R} \times \mathcal{G}$, and there is $\tau \in (0, 1]$ such that

$$r(z \mid x) \geq \tau, \qquad g(y \mid x, z) \geq \tau$$

for all $r \in \mathcal{R}$, $g \in \mathcal{G}$, and all x, y, z. If

$$n \geq \frac{2(1+\beta)^2\log^2(1/\tau)}{(1-\beta)^2(\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta))^2}\log\frac{2|\mathcal{Q}||\mathcal{R}||\mathcal{G}|}{\eta},$$

then with probability at least $1-\eta$, every empirical profiled-ELBO minimizer satisfies $d_\rho(\widehat{q}_n, S^\star) < \delta$.

Proof. For a fixed triple (q, r, g), the data-dependent part of the single-sample loss is

$$\sum_z q(z \mid x, y) \left[-\log g(y \mid x, z) + \beta \log \frac{q(z \mid x, y)}{r(z \mid x)} \right].$$

The support floor gives $-\log g(y \mid x, z) \le \log(1/\tau)$, and

$$\sum_z q(z \mid x, y) \log \frac{q(z \mid x, y)}{r(z \mid x)} = \mathrm{KL}(q(\cdot \mid x, y) \| r(\cdot \mid x)) \le \log \frac{1}{\tau}.$$

Thus the data-dependent part is bounded between 0 and $(1 + \beta) \log(1/\tau)$. The complexity term $\lambda C(r, g)$ is deterministic and cancels in empirical-population deviations.

Hoeffding's inequality and a union bound over $\mathcal{Q} \times \mathcal{R} \times \mathcal{G}$ imply

$$\mathbb{P}\left(\sup_{q,r,g} \left| \widehat{\mathcal{L}}_{\beta,\lambda,n}(q, r, g) - \mathcal{L}_{\beta,\lambda}(q, r, g) \right| \ge t \right) \le 2|\mathcal{Q}||\mathcal{R}||\mathcal{G}| \exp\left(-\frac{2nt^2}{(1 + \beta)^2 \log^2(1/\tau)} \right).$$

Set

$$t = \frac{(1 - \beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)}{2}.$$

The displayed lower bound on n makes the right-hand side at most η.

On the complementary event, profiling cannot increase the deviation: for every q,

$$\left| \inf_{r,g} \widehat{\mathcal{L}}_{\beta,\lambda,n}(q, r, g) - \inf_{r,g} \mathcal{L}_{\beta,\lambda}(q, r, g) \right| \le \sup_{r,g} \left| \widehat{\mathcal{L}}_{\beta,\lambda,n}(q, r, g) - \mathcal{L}_{\beta,\lambda}(q, r, g) \right|.$$

Therefore

$$\sup_{q \in \mathcal{Q}} \left| \widehat{\overline{\mathcal{L}}}_{\beta,\lambda,n}(q) - \overline{\mathcal{L}}_{\beta,\lambda}(q) \right| < \frac{(1 - \beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)}{2}.$$

The empirical recovery theorem then gives $d_\rho(\widehat{q}_n, S^\star) < \delta$ for every empirical profiled-ELBO minimizer. □

Support floors are only one transparent sufficient condition. Other standard routes are to clip predictive probabilities, assume direct concentration of the profiled losses, prove a high-probability envelope or sub-exponential tail bound for the relevant log losses, or use entropy, bracketing, or Rademacher control for the likelihood class. The recovery argument itself only needs uniform accuracy at scale $(1 - \beta)\Delta^{\rho,\mathrm{prof}}_{\beta,\lambda}(\delta)$; support or tail assumptions are one way to obtain that event.