

Token-Level Control or Just a Better Mean? Isolating the Gains of Adaptive Decoding

Qingyun Li, Ruoyun Li

Paul G. Allen School of Computer Science and Engineering
University of Washington
{qxli2, rxli2}@uw.edu

Abstract

Adaptive decoding methods that predict per-token sampling parameters have reported substantial gains over fixed decoding strategies. But what drives these gains: genuine token-level control, or simply a better average configuration? We develop an *inverse-temperature decomposition* that cleanly separates two mechanisms: shifting the global operating point and performing residual token-level adaptation. The theory shows that teacher-forced log-probability is strictly concave in inverse temperature, so any token-level variance incurs a quadratic curvature penalty that must be overcome by precise gradient alignment. Testing this framework on representative learned and heuristic controllers across three model families and four benchmarks spanning reasoning and knowledge tasks reveals a striking pattern: a fixed operating point (obtained by simply averaging the learned controller’s own predictions) matches or exceeds the learned controller (AUTODECO) in the large majority of evaluated settings. Mechanistic diagnostics confirm the theory: residual temperature variation exhibits near-chance gradient alignment and low Cauchy–Schwarz efficiency, with curvature penalties dominating on the uncertain tokens where adaptive control would matter most. A complementary analysis suggests similar structural limits for adaptive top- p . These findings trace to the limits of low-dimensional sampling control rather than to any one controller. The decomposition toolkit and MEANSHIFT baseline we introduce give the field a sharper test of genuine token-level adaptation.

1 Introduction

Language model sampling involves a continuous exploration–exploitation trade-off. Lower temperatures and restrictive top- p thresholds favor precision (useful for reasoning, mathematics, and code) while higher temperatures encourage the diversity needed for coverage and creative generation. Since the optimal configuration varies across tasks, prompts, and even individual tokens, *adaptive decoding* (adjusting sampling parameters dynamically during generation) is a natural and appealing idea. Recent work on adaptive decoding includes both learned hidden-state-conditioned controllers and hand-designed heuristic rules. AdaptiveDecoder/LPO (Dhuliawala et al., 2024) predicts temperature at the token or sequence level from the last hidden state via a learned distribution over discrete temperature choices, while AUTODECO (Wang et al., 2026) extends this interface to joint per-token prediction of temperature and top- p . Other approaches use output-distribution statistics to modulate decoding parameters heuristically (Zhu et al., 2024; 2023). Together, these methods present a compelling case for dynamic decoding hyperparameters.

The conflation problem. Current evaluations do not separate two fundamentally different mechanisms through which an adaptive controller can improve performance:

1. **Operating-point correction.** The model’s default decoding parameters may simply be suboptimal; the controller’s *average* predicted temperature and top- p may constitute a better static configuration.

2. **Token-level adaptation.** The controller provides genuinely useful per-token modulation that a fixed configuration cannot replicate.

Every published comparison we are aware of tests adaptive decoding against a default fixed configuration, so any gain could be either effect or both. This is not a minor methodological gap: it determines whether the field is making progress on a hard and interesting problem (learning when to be stochastic) or rediscovering that default decoding parameters are not optimal.

Our approach. We introduce an *inverse-temperature decomposition* that makes this separation precise. Working in inverse-temperature space ($\beta = 1/T$), where teacher-forced log-probability is strictly concave, we decompose any adaptive controller into a fixed operating-point component $\bar{\beta}$ and a token-level residual δ_t . The theory shows that residual variation incurs a quadratic *curvature penalty* proportional to logit variance, which must be offset by precise *gradient alignment* between the residual and the tokenwise slope. This yields a concrete prediction: if alignment is weak, a fixed operating point equal to the controller’s mean prediction (which we call MEANSHIFT) should match or exceed the full adaptive controller.

Key findings.

- **Operating-point dominance.** MEANSHIFT matches or exceeds the learned controller (AUTODECO) on the large majority of benchmark–metric combinations across three model families (7B–30B, dense and MoE), so the dominant effect is a better average configuration rather than token-level control (Section 5).
- **Weak alignment.** Residual temperature variation has a small positive local effect under teacher forcing, but the correlation between the residual and the tokenwise gradient is $r \approx 0.05$ with sign agreement indistinguishable from chance (Section 6.1).
- **Adverse leverage localization.** Confident tokens offer negligible headroom for any adaptive controller. Uncertain tokens offer high leverage, but the curvature penalty exceeds the alignment gain in those regions (Section 6.2).
- **Top- p is also structurally limited.** A complementary appendix analysis suggests that adaptive top- p acts mainly through sparse support changes rather than smooth control (Appendix H).

Scope and contribution. We study the prevailing interface for adaptive decoding: post-hoc control of a frozen next-token distribution through a small number of sampling parameters, instantiated here by temperature and top- p . We use AUTODECO as our primary learned case study because it is the strongest public method on this interface, but our goal is not to criticize AUTODECO specifically. Rather, we identify structural limits of low-dimensional sampling control and introduce a diagnostic framework (the decomposition, the MEANSHIFT baseline, and the alignment metrics) that we argue should become a standard evaluation component for future adaptive decoding work. Expanded related work, including broader adaptive inference-time methods with different control surfaces, appears in Appendix A.1.

We stress that MEANSHIFT is a *matched projection test*, not an upper bound on token-level control. Given any adaptive controller, it preserves the operating point the controller discovers and removes only its token-level variation. A controller that beats its own projection has shown that its residual policy does useful work; one that does not has gains that are difficult to attribute to token-level control rather than to the operating point.

2 Background and Setup

2.1 Notation

We focus on temperature and top- p , the primary stochasticity controls in practice and the control surface of the methods we study.

Let $h_t \in \mathbb{R}^d$ denote the final hidden state at decoding step t , and let $z_t = W_{\text{lm}} h_t \in \mathbb{R}^{|V|}$ be the pre-softmax logits over vocabulary V . A standard stochastic decoder first forms a temperature-scaled distribution

$$q_t(i; T) = \text{softmax}\left(\frac{z_t}{T}\right)_i, \quad (1)$$

and may then apply an additional filtering or truncation rule, such as top- p sampling, before drawing the next token.

In the control family studied here, adaptive decoding replaces a fixed temperature with token-dependent T_t , and in some methods also replaces fixed top- p with token-dependent p_t . Learned controllers predict these quantities from h_t (Dhuliawala et al., 2024; Wang et al., 2026); heuristic controllers derive them from statistics of z_t , such as entropy. For fixed logits, temperature and predictive entropy are in one-to-one correspondence, since $H(\text{softmax}(z_t/T))$ is strictly increasing in T (Appendix C). Thus entropy-based adaptive rules can also be interpreted as reparameterized temperature controllers rather than a separate control family.

2.2 A hierarchy of baselines

To isolate the source of adaptive decoding gains, we define a strict hierarchy of comparisons:

1. **Greedy decoding.** The deterministic strategy that always selects the highest-probability token at each generation step.
2. **Base sampling.** The default static T and top- p provided by standard model inference.
3. **Grid search.** A fixed (T, p) selected by coarse grid search on a small development set, without access to the learned controller (Appendix K). If this recovers the same operating region as MEANSHIFT, the learned controller is not needed even to identify the right fixed point.
4. **MEANSHIFT.** A static strategy that fixes T and top- p to the empirical mean of the adaptive controller’s predictions on a held-out corpus (see Appendix D for estimation details). This isolates operating-point correction: any gain of MEANSHIFT over base sampling is attributable to a better average configuration, with no token-level variation.
5. **Adaptive controllers (learned or heuristic).** Methods that vary T_t and p_t per token. If these do not substantially exceed MEANSHIFT, then their dominant effect is operating-point correction.

3 Theory: An Inverse-Temperature Decomposition

We now develop the theoretical framework that motivates and predicts our empirical findings. The argument proceeds in three steps: (i) the teacher-forced log-probability is strictly concave in inverse temperature, creating a structural penalty for token-level variance; (ii) decomposing the controller into a mean and a residual makes this penalty explicit; and (iii) achieving the alignment needed to overcome this penalty is difficult in that the tokens that offer the most leverage for residual control are precisely the tokens where the curvature penalty is largest.

3.1 Concavity in inverse temperature

Define the inverse temperature $\beta = 1/T$. The teacher-forced log-probability of the gold token y_t at step t is

$$\ell_t(\beta) = \log \text{softmax}(\beta z_t)_{y_t} = \beta z_{t,y_t} - \log \sum_j e^{\beta z_{t,j}}. \quad (2)$$

Proposition 1 (Strict concavity). *For logits z_t with at least two distinct entries, $\ell_t(\beta)$ is strictly concave in β on $(0, \infty)$:*

$$\ell_t''(\beta) = -\text{Var}_{q_t(\beta)}[z_{t,j}] < 0, \quad (3)$$

where $q_t(\beta) = \text{softmax}(\beta z_t)$.

This is the standard concavity of the log-partition function in its natural parameter (Boyd & Vandenberghe, 2004), specialized to the softmax; the proof follows by direct differentiation (Appendix B). Strict concavity means that *any* token-level variance in β around the optimum incurs a cost, by Jensen’s inequality, an exact structural property of the softmax rather than an approximation.

3.2 Mean-residual decomposition

Write the adaptive controller’s inverse-temperature prediction as

$$\beta_t = \bar{\beta} + \delta_t, \quad (4)$$

where $\bar{\beta} = \mathbb{E}[\beta_t]$ is the mean inverse temperature (the global operating point) and δ_t is the token-level residual. A second-order Taylor expansion around $\bar{\beta}$ gives:

$$\ell_t(\bar{\beta} + \delta_t) - \ell_t(\bar{\beta}) \approx \underbrace{g_t \delta_t}_{\text{alignment gain}} - \underbrace{\frac{1}{2} v_t \delta_t^2}_{\text{curvature penalty}}, \quad (5)$$

where $g_t = \ell'_t(\bar{\beta})$ is the local gradient and $v_t = -\ell''_t(\bar{\beta}) = \text{Var}_{q_t(\bar{\beta})}[z_{t,j}] > 0$ is the local curvature.

Interpretation. The alignment gain $g_t \delta_t$ is positive when the controller moves β in the direction of the local gradient: increasing β (lowering T) when the gold token’s logit exceeds the probability-weighted mean, and decreasing β otherwise. The curvature penalty $\frac{1}{2} v_t \delta_t^2$ is always nonnegative and grows quadratically in the residual’s magnitude. For the residual to help on average, the expected alignment must exceed the expected penalty:

$$\mathbb{E}[g_t \delta_t] > \frac{1}{2} \mathbb{E}[v_t \delta_t^2]. \quad (6)$$

3.3 Where alignment helps most, curvature costs most

The gradient at token t is

$$g_t = \ell'_t(\bar{\beta}) = z_{t,y_t} - \mathbb{E}_{q_t(\bar{\beta})}[z_{t,j}], \quad (7)$$

i.e., the difference between the gold token’s logit and the probability-weighted mean logit. To achieve strong alignment, the controller must predict not merely *whether* the current token is uncertain (which simple statistics like entropy capture well), but also where the gold token sits *relative to the distribution’s center*, which is a subtler and more volatile signal.

Moreover, the tokens where adaptive control matters most (those with high entropy and broad probability mass) are precisely the tokens where the curvature $v_t = \text{Var}_{q_t}[z_{t,j}]$ is largest. This creates a structural trap:

- On **confident tokens** ($p_{\max}$ high, entropy low), the logit variance is small (so the curvature penalty is small), but so is the potential alignment gain. There is little headroom for any adaptive controller.
- On **uncertain tokens** ($p_{\max}$ low, entropy high), the potential alignment gain is large, but the curvature penalty is severe, and the gradient g_t swings widely, making it the hardest to predict correctly.

The optimal residual that maximizes Equation 5 per token is $\delta_t^* = g_t / v_t$, yielding a maximum possible per-token gain of $g_t^2 / (2v_t)$. While the local curvature v_t determines the optimal *magnitude* of the adjustment, the gradient g_t dictates its *direction*. Because the quadratic curvature penalty applies to any temperature variation, failing to align with g_t guarantees a net loss, regardless of how cleverly the controller scales its moves relative to v_t . Realizing even a fraction of this theoretical gain therefore requires that the controller’s residual correlate strongly with g_t ; this is a demanding requirement that, as we show empirically, current controllers do not meet.

3.4 Empirical hypotheses

The decomposition yields two testable predictions:

Hypothesis 1: Operating-point dominance. If the alignment between δ_t and g_t is weak, then token-level variation will provide minimal benefit beyond the mean operating point, and MEANSHIFT should be competitive with or better than the full adaptive controller.

Hypothesis 2: Adverse leverage localization. The controller should be most *harmful* on the tokens where it matters most. On high-entropy tokens, where the potential gain from correct adaptation is largest, the curvature penalty should exceed the alignment gain (so the residual actively hurts relative to the mean operating point). On low-entropy tokens, both terms are small and the controller should have little leverage in either direction.

4 Experimental Setup

4.1 Evaluated methods

In the main text, we compare methods that act on the same post-hoc sampling-parameter interface. Our primary learned method is AUTODECO (Wang et al., 2026), which uses two lightweight heads over hidden states to predict token-level temperature and top- p ; we use the publicly released weights for each model. As heuristic comparators, we include EDT (Zhang et al., 2024) (a prior method on this interface), and two simple reference baselines that we construct for this study: CONFGATE and ENTROPYSHIFT. Brief definitions and tuned hyperparameters for these heuristics appear in Appendix D.4. Related adaptive inference methods that intervene through different mechanisms are discussed in Appendix A.1, since they are informative context but not directly matched baselines for the decomposition studied here.

4.2 Models

Primary experiments use **DeepSeek R1 Distill Qwen 7B**. To test generalization across model families, we repeat the full evaluation on two additional open-weight models: **Qwen3-30B-A3B-Instruct-2507** (a mixture-of-experts instruction-tuned model) and **GPT-OSS-20B** (a mixture-of-experts reasoning model). These three models span different sizes (7B–30B), architectures (dense vs. MoE), and training pipelines, providing a meaningful test of whether operating-point dominance is model-specific or structural. Cross-model results are summarized in Section 5.3; detailed per-benchmark tables appear in Appendix J.

For each model, **Base** decoding uses that model family’s recommended default decoding configuration; exact values are listed in Appendix D.

4.3 Benchmarks

AUTODECO is trained on reject-sampled reasoning trajectories from DeepMath-103k (He et al., 2026). As such, for our main-paper results, we report four benchmarks spanning matched-domain and out-of-domain evaluation: AIME24 and HMMT25 (Balunović et al., 2025) are competition-style math benchmarks; GPQA-Diamond (Rein et al., 2024) and MMLU-Pro (Wang et al., 2024) are science/knowledge benchmarks. For MMLU-Pro, we follow Wang et al. (2026) and use a 100-question “lite” subset¹. Appendix E further tests whether the same qualitative pattern extends to more open-ended, generation-style instruction-following benchmarks, using IFEval (Zhou et al., 2023) and IFBench (Pyatkin et al., 2025).

4.4 Metrics

We generate $k = 16$ samples per problem and report:

¹<https://huggingface.co/datasets/koiwave/100MMLUpro>

Benchmark	Metric	Base	Greedy	AUTODECO	MEANSHIFT	Δ
AIME24	maj@1	52.97 $\pm$ 1.14	50.00 $\pm$ 0.00	53.44 $\pm$ 1.46	55.91$\pm$1.01	+2.47
	maj@16	49.17 $\pm$ 4.15	50.00 $\pm$ 0.00	52.50 $\pm$ 3.25	55.42$\pm$1.44	+2.92
	pass@16	83.75$\pm$2.33	50.00 $\pm$ 0.00	78.75 $\pm$ 1.44	81.25 $\pm$ 1.44	+2.50
GPQA-Diamond	maj@1	47.94 $\pm$ 0.55	36.87 $\pm$ 0.00	48.72 $\pm$ 0.35	49.49$\pm$0.39	+0.77
	maj@16	42.99 $\pm$ 0.66	36.87 $\pm$ 0.00	43.31 $\pm$ 1.35	44.19$\pm$0.98	+0.88
	pass@16	89.02 $\pm$ 1.33	36.87 $\pm$ 0.00	90.34 $\pm$ 0.48	91.79$\pm$1.80	+1.45
HMMT25	maj@1	24.02 $\pm$ 0.55	23.33 $\pm$ 0.00	23.68 $\pm$ 0.47	24.47$\pm$0.69	+0.79
	maj@16	18.12 $\pm$ 1.12	23.33$\pm$0.00	18.54 $\pm$ 1.12	20.21 $\pm$ 1.37	+1.67
	pass@16	58.54$\pm$1.71	23.33 $\pm$ 0.00	55.42 $\pm$ 2.58	58.33 $\pm$ 3.11	+2.91
MMLU-Pro-lite	maj@1	42.66 $\pm$ 0.45	35.00 $\pm$ 0.00	43.45 $\pm$ 0.40	43.58$\pm$0.49	+0.13
	maj@16	38.38 $\pm$ 1.09	35.00 $\pm$ 0.00	40.38$\pm$1.26	39.62 $\pm$ 0.77	-0.76
	pass@16	78.12 $\pm$ 1.58	35.00 $\pm$ 0.00	80.50$\pm$1.84	79.26 $\pm$ 1.55	-1.24

Table 1: Main benchmark results on four evaluation benchmarks (mean $\pm$ 95% CI across seeds; higher is better). The final column reports $\Delta = \text{MEANSHIFT} - \text{AUTODECO}$ in percentage points. **Bold** = best mean, underline = second best.

- **maj@1**: expected single-sample accuracy (the natural “one-shot” metric).
- **maj@k**: whether the majority answer among k samples is correct (exploitation).
- **pass@k**: whether at least one of k samples reaches a correct final answer (exploration).

We follow Wang et al. (2026) and run 8 random seeds per benchmark. For each metric, we report mean $\pm$ 95% CI (details provided in Appendix D.6).

4.5 Protocol

MEANSHIFT fixes T and p to the empirical mean of AUTODECO’s tokenwise predictions, estimated from approximately 1M free-generation tokens drawn from a held-out auxiliary reasoning corpus (Appendix D.1 and Appendix D.2). The teacher-forced analyses reported in Section 6 are computed on a separate held-out sample from the same auxiliary corpus. For heuristic methods, we lightly tune hyperparameters on a small held-out development set and then fix the selected setting across all reported evaluation benchmarks (Appendix D.4); MEANSHIFT is not tuned on this set.

5 Main Results: Operating-Point Correction Dominates

5.1 MEANSHIFT versus learned control

The central empirical test is straightforward: if our theory is correct, MEANSHIFT should be competitive with or better than the learned controller AUTODECO.

Table 1 shows that **MEANSHIFT matches or exceeds AUTODECO on all three metrics for AIME24, GPQA-Diamond, and HMMT25**, and on maj@1 for MMLU-Pro-lite. AUTODECO retains a slight advantage on MMLU-Pro-lite maj@16 and pass@16 (so the learned controller is not uniformly dominated) but the overall picture is clear. We also show in Appendix F that this qualitative pattern persists across sample budgets.

This directly confirms Hypothesis 1: both AUTODECO and MEANSHIFT show improvements over Base decoding (confirming that the controller successfully discovers a better static operating point) but the residual token-level variation adds no consistent value *beyond that shift* and sometimes mildly hurts, exactly as our curvature–alignment trade-off predicts. Section 5.3 shows that the same qualitative pattern holds across the two additional model families we evaluate.

Independence from the learned controller. A coarse 3×2 grid search over (T, p) on the development set (Appendix K) independently selects (0.8, 0.95)—close to MEANSHIFT’s

Benchmark	Metric	CONFGATE	ENTROPYSHIFT	EDT	MEANSHIFT
AIME24	maj@1	49.01±6.52	53.91±0.82	54.66±1.55	55.91±1.01
	maj@16	46.67±7.60	53.33±3.65	55.00±5.37	55.42±1.44
	pass@16	76.25±9.47	81.67±2.98	81.25±2.07	<u>81.25±1.44</u>
GPQA-Diamond	maj@1	49.03±0.39	49.08±0.51	48.96±0.47	49.49±0.39
	maj@16	43.81±1.21	43.31±0.70	44.16±0.75	44.19±0.98
	pass@16	90.15±1.11	90.21±1.24	<u>90.66±1.56</u>	91.79±1.80
HMMT25	maj@1	24.15±0.83	24.05±0.45	24.27±0.76	24.47±0.69
	maj@16	19.58±1.70	19.76±1.33	18.96±1.41	20.21±1.37
	pass@16	57.71±2.81	53.33±2.83	<u>57.29±3.32</u>	58.33±3.11
MMLU-Pro-lite	maj@1	42.32±0.48	42.85±0.45	<u>43.15±0.45</u>	43.58±0.49
	maj@16	38.62±1.48	38.95±1.15	<u>39.25±0.95</u>	39.62±0.77
	pass@16	77.88±1.37	78.45±1.45	<u>78.72±1.35</u>	79.26±1.55

Table 2: Heuristic adaptive baselines versus MEANSHIFT (mean $\pm$ 95% CI across seeds). MEANSHIFT is best on 11/12 rows; ENTROPYSHIFT exceeds it only on AIME24 pass@16. **Bold** = best mean, underline = second best.

(0.798, 0.907)—and matches its evaluation performance across all four benchmarks (≤ 1 pp difference on every metric; Table 15). The operating-point landscape is relatively smooth and broad, so the controller’s primary function is locating a region that a trivial sweep discovers without any access to the learned controller.

5.2 Heuristic adaptive baselines

If operating-point correction is the main effect, do heuristic adaptive methods add value beyond a strong fixed point? Table 2 shows that, in most cases, they do not.

EDT, the strongest heuristic method overall, is notable because it was specifically designed to modulate temperature as a function of output entropy (Zhang et al., 2024), precisely the kind of signal our theory identifies as necessary but insufficient. ENTROPYSHIFT and EDT occasionally approach or match MEANSHIFT on individual metrics, but neither systematically exceeds it across benchmarks. ENTROPYSHIFT is anchored so that its average shift is zero at the MEANSHIFT operating point (Appendix D.4), so it is the strictest of these comparisons by construction (it competes on entropy-conditioned token-level variation alone). That it still does not exceed MEANSHIFT isolates the residual as the component that fails. Heuristic controllers facing the same barrier as the learned controller is the strongest evidence for our central claim: the difficulty is intrinsic to the *structure of token-level sampling parameter control*, and this applies regardless of whether the residual is produced by a neural network or a hand-designed rule.

5.3 Cross-model generalization

We repeat the full AUTODECO–MEANSHIFT comparison on two additional model families to test whether operating-point dominance is specific to DeepSeek-R1-Distill-Qwen-7B or a broader property of the control interface. Table 3 summarizes the results; full per-benchmark tables are given in Appendix J.

The pattern is directionally consistent across all three models (29/36 cells, positive macro-average Δ on each metric), with the effect strongest on maj@1 and maj@16 and somewhat weaker on pass@16, where diversity rather than single-sample accuracy matters more (Appendix F). This consistency across model families is what we would expect if the barrier to token-level control is a property of softmax curvature and the low-dimensional sampling interface rather than an artifact of any single model.

Model	Δ maj@1	Δ maj@16	Δ pass@16	Mean Δ	Wins
DeepSeek-R1-Distill-Qwen-7B	+1.04	+1.18	+1.41	+1.21	10/12
Qwen3-30B-A3B-Instruct-2507	+1.02	+0.55	-0.28	+0.43	9/12
GPT-OSS-20B (medium)	+0.94	+1.52	+0.53	+0.99	10/12
Macro average	+1.00	+1.08	+0.55	+0.88	29/36

Table 3: Cross-model generalization summary. Δ denotes MEANSHIFT – AUTODECO (benchmark-macro-averaged for the listed metric); Mean Δ averages over all 12 benchmark-metric cells, and Wins counts the number of cells where MEANSHIFT equals or exceeds AUTODECO. Operating-point dominance persists across all three model families.

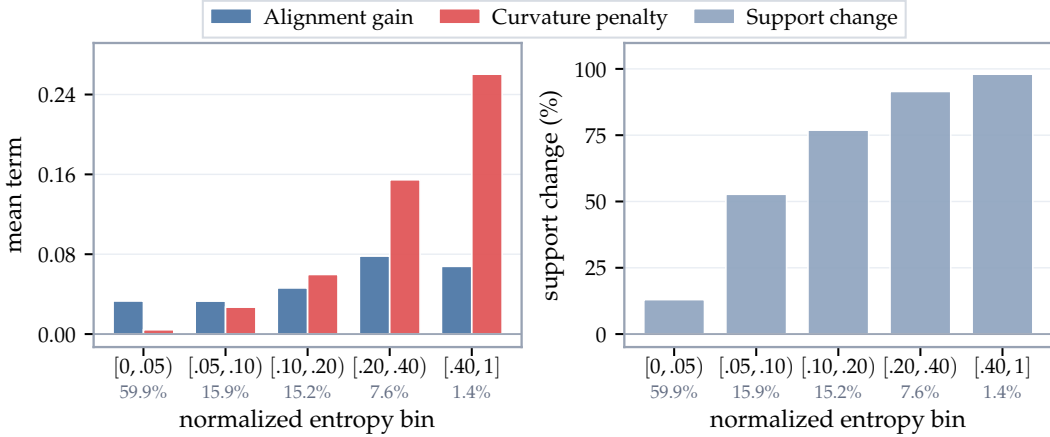


Figure 1: **Left:** Leverage localization by normalized entropy (token distribution across bins is shown below the x-axis). Alignment gain rises on uncertain tokens, but curvature penalty rises faster. **Right:** Around 60% of tokens are low-entropy, but higher-entropy bins are where most support changes happen.

6 Mechanistic Diagnostics

The main results establish *that* operating-point correction dominates. This section explains *why*, using the inverse-temperature decomposition to probe the learned controller’s residual behavior on a separate held-out teacher-forced diagnostic corpus.

6.1 Inverse-temperature alignment test

We compute the decomposition from Equation 5 for AUTODECO’s residual $\delta_t = \beta_t - \hat{\beta}$. As Table 4 shows, the residual exhibits weak alignment with the tokenwise gradient: correlation is small, sign agreement is near chance, and Cauchy–Schwarz efficiency is low. Correspondingly, the curvature penalty dominates on most tokens, and the residual improves gold-token log-probability on fewer than 40% of them. Although the residual does carry a small net positive signal (Appendix G.1), it is too weak to overcome the structural penalty identified in Section 3.

6.2 Leverage localization

Where does the controller attempt to exert control, and where does it pay the highest price? We bin tokens by normalized entropy $\hat{H}_t = H_t / \log |V|$ and measure the alignment gain and curvature penalty in each bin.

The pattern (Figure 1) directly confirms Hypothesis 2. Most tokens are confident: 59.9% fall in the $[0.00, 0.05)$ entropy bin, where the alignment gain is 0.033 and the penalty is only 0.004. As entropy rises, curvature penalty increases sharply; by the $[0.10, 0.20)$ bin, the

Metric	Value	
$\text{Corr}(g, \delta)$	0.052	near-zero gradient alignment
Sign agreement	49.1%	indistinguishable from chance
Cauchy–Schwarz Efficiency	5.66%	residual nearly orthogonal to gradient
Penalty > gain	63.3%	fraction of tokens where penalty dominates
Frac. positive gain	39.8%	fraction of tokens where δ_t helps

Table 4: Inverse-temperature alignment diagnostics for AUTODECO. All three alignment measures are near chance; the curvature penalty dominates on a majority of tokens. Full decomposition components and formal definitions are given in Appendix G.1; the second-order Taylor approximation tracks the true per-token gain with Pearson $r = 0.991$.

penalty already exceeds the alignment gain (0.060 vs. 0.046). In the $[0.20, 0.40)$ bin, which still contains 7.6% of tokens, the penalty (0.154) is nearly twice the alignment gain (0.079), and support changes affect 91.5% of tokens.

This shows that the difficulty of token-level control is not a matter of insufficient training data or model capacity. It is a structural consequence of the relationship between uncertainty and curvature. The tokens where adaptive control would help most are precisely the tokens where it is hardest to get right.

Analysis on top- p . Proposition 1 and the decomposition that follows from it apply to temperature control only. Top- p has a different mathematical structure; it changes the sampling support at discrete cumulative-mass breakpoints rather than smoothly reweighting the full distribution, so no analogous curvature argument is available and we do not claim one. Our treatment of top- p is correspondingly empirical and is deferred to Appendix H. In brief, adaptive top- p changes the sampling support on a substantial minority of tokens, but these changes are concentrated near breakpoints rather than reflecting smooth token-level control (a different mechanism that supports the same conclusion).

6.3 Teacher forcing versus free generation

The diagnostics above are computed under teacher forcing, while the results in Section 5 are free-generation evaluations. We use free generation to establish the performance comparison and teacher forcing to isolate the local mechanism behind it. To verify that the diagnostic probes the same controller behavior seen during generation, we recompute AUTODECO’s operating statistics in both regimes over approximately 1M tokens each.

Regime	T	p	$\hat{H}$	$p_{\max}$	Logit var.
Free generation	0.798±0.208	0.907±0.105	0.072±0.103	0.786±0.264	1.933±3.103
Teacher-forced	0.769±0.184	0.915±0.107	0.039±0.073	0.874±0.215	1.001±2.138

Table 5: AUTODECO operating statistics under teacher forcing and free generation on DeepSeek-R1-Distill-Qwen-7B (mean $\pm$ SD over approximately 1M tokens per regime). T and p are the controller’s predicted sampling parameters; $\hat{H}$ is normalized next-token entropy; $p_{\max}$ is the probability of the most likely next token.

The controller’s average operating point is nearly identical across regimes (T : 0.798 vs. 0.769; p : 0.907 vs. 0.915). What differs is the distribution it acts on: teacher-forced contexts are more peaked and lower entropy, with $p_{\max}$ of 0.874 against 0.786 under free generation and normalized entropy of 0.039 against 0.072. This sharpens the reading of the diagnostic in both directions. Teacher forcing gives residual control a clean local opportunity, since the gold-token direction is known and trajectory compounding is removed, and even there, on these more confident distributions, the residual is weakly aligned and the curvature penalty dominates. Free generation places the same controller at nearly the same operating point on somewhat higher-entropy contexts, but Section 6.2 shows the added headroom sits

in exactly the high-entropy regions where the curvature penalty is most severe. The two regimes therefore point the same way as the benchmark results.

7 Discussion

What we claim. Our results show that, across three model families and four benchmarks, the dominant mechanism behind current adaptive-decoding gains through low-dimensional sampling parameters (temperature and top- p) is operating-point correction. We are not arguing that token-level adaptation is inherently impossible, but rather that the current interface of low-dimensional post-hoc sampling control over frozen logits does not provide enough leverage to overcome the structural penalty we identify.

When might learned token-level control help? A stronger token-level controller should both outperform a matched fixed operating point and show materially better alignment with the tokenwise gradient than we observe here, e.g., higher $\text{Corr}(g, \delta)$, sign agreement well above chance, and substantially larger Cauchy–Schwarz efficiency. Future adaptive decoding work should therefore report these diagnostics together with MEANSHIFT (or a more general static projection; Appendix D.3), which clarifies whether the gains reflect genuine token-level control or primarily operating-point discovery. Plausible paths to stronger alignment include controllers trained jointly with model adaptation (rather than post-hoc), control surfaces richer than temperature and top- p (e.g., direct logit intervention), architectures whose control signal depends on long-range structural properties invisible to local logit statistics, and problem domains where the optimal temperature varies more dramatically across tokens than in the benchmarks we study.

Limitations. Our main-paper evaluation focuses on reasoning and knowledge tasks; appendix results on instruction-following benchmarks provide a preliminary extension, but we do not evaluate creative or long-form generation. At the same time, this is a favorable test bed for AUTODECO, since it includes the regime most closely aligned with its training and original evaluation setup. Our implementation of MEANSHIFT uses the empirical coordinatewise means of temperature and top- p ; this is a good fit for the models we test, but more complex or multimodal adaptive policies may be better compared against a generalized static projection rather than a raw parameter mean (Appendix D.3). The inverse-temperature decomposition also remains a teacher-forced analysis of temperature control. Section 6.3 shows the controller operates at a nearly identical average point in both regimes, with free generation offering less headroom for residual variation, but this connects the two only in aggregate; future work could explore modeling trajectory-level effects such as distributional shift or error propagation, as well as the impact of the sampling budget k .

8 Conclusion

We investigated whether adaptive decoding gains come from genuine token-level control or from a better average configuration. An inverse-temperature decomposition shows that variance in decoding temperature incurs a quadratic curvature penalty, so token-level adaptation must achieve strong gradient alignment to be worthwhile. Across reasoning and knowledge benchmarks and three model families, a fixed operating point derived from the controller’s own mean predictions consistently matches or exceeds token-level control. Diagnostics explain why: residual temperature variation is weakly aligned and penalties dominate where adaptation would matter most. Overall, the dominant gain of current adaptive decoding is operating-point correction. We hope the decomposition, diagnostics, and MEANSHIFT baseline introduced here help the field build adaptive methods that genuinely surpass their own average operating point.

Ethics Statement

This work studies decoding-time control for language models. Potential benefits include more transparent and better-grounded evaluation of generation strategies. Potential risks

include overgeneralizing from a limited set of models or benchmarks, especially in domains where decoding choices interact with safety or trust. We report negative as well as positive results and encourage careful evaluation before deployment in high-stakes settings.

Reproducibility Statement

Code for all baselines, the MEANSHIFT estimation procedure, the inverse-temperature diagnostics, and every evaluation script is available at <https://github.com/cminst/SimpleDeco>. The release includes the fixed MEANSHIFT operating points used for each model (Table 6), the heuristic-baseline hyperparameters selected on the development set, and the per-seed benchmark outputs underlying every table in this paper. The appendix documents model versions and checkpoints (Appendix D.2), the auxiliary corpus and sampling protocol used for operating-point estimation and diagnostics (Appendices D.1 and G.3), benchmark settings and answer-extraction rules (Appendix D.5), development-set tuning procedures (Appendix D.4), and random seeds and confidence-interval computation (Appendix D.6).

References

- Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating LLMs on uncontaminated math competitions, February 2025. URL <https://matharena.ai/>.
- Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *International Conference on Learning Representations*, 2021.
- Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- Shehzaad Dhuliawala, Ilia Kulikov, Ping Yu, Asli Celikyilmaz, Jason Weston, Sainbayar Sukhbaatar, and Jack Lanchantin. Adaptive decoding via latent preference optimization. *arXiv preprint arXiv:2411.09661*, 2024.
- Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 101–109, 2019.
- Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.
- Kaifeng He, Mingwei Liu, Chong Wang, Zike Li, Yanlin Wang, Xin Peng, and Zibin Zheng. Towards better code generation: Adaptive decoding with uncertainty guidance. *arXiv preprint arXiv:2506.08980*, 2025.
- Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations (ICLR)*, 2026. doi: 10.48550/arXiv.2504.11456.
- John Hewitt, Christopher D. Manning, and Percy Liang. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3414–3427, 2022.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. doi: 10.48550/arXiv.2106.09685.

- Kuan-Wei Lu, Ding-Yong Hong, and Pangfeng Liu. Adasd: Adaptive speculative decoding for efficient language model inference. *arXiv preprint*, 2025.
- Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? a large-scale study. In *Advances in Neural Information Processing Systems*, volume 31, pp. 700–709, 2018.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023.
- Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*, 2018.
- Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G. Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *International Conference on Learning Representations*, 2025.
- Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision – ECCV 2020*, volume 12370 of *Lecture Notes in Computer Science*, pp. 681–699, 2020.
- Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Thirty-Ninth Conference on Neural Information Processing Systems (NeurIPS 2025), Datasets and Benchmarks Track*, 2025. doi: 10.48550/arXiv.2507.02833.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling (COLM)*, 2024. doi: 10.48550/arXiv.2311.12022.
- Huangyuan Su, Zhe Ye, Samuel Tenka, Aidan Z. H. Yang, Soonho Kong, and Udaya Ghai. Learning adaptive LLM decoding. In *ICML 2026 Workshop on Resource-Adaptive Foundation Model Inference (AdaptFM)*, 2026.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290, 2024. doi: 10.52202/079017-3018.
- Zhichao Wang, Dongyang Ma, Xinting Huang, Deng Cai, Tian Lan, Jiahao Xu, Haitao Mi, Xiaoying Tang, and Yan Wang. The end of manual decoding: Towards truly end-to-end language models. In *International Conference on Learning Representations (ICLR)*, 2026. doi: 10.48550/arXiv.2510.26697.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Shimao Zhang, Yu Bao, and Shujian Huang. EDT: Improving large language models’ generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*, 2024.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.
- Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. Improving open-ended text generation via adaptive decoding. *arXiv preprint arXiv:2402.18223*, 2024.
- Yifan Zhu, Huiqiang Rong, and Haoran Luo. Token-guard: Towards token-level hallucination control via self-checking decoding. In *International Conference on Learning Representations (ICLR)*, 2026. doi: 10.48550/arXiv.2601.21969.

Yuqi Zhu, Jia Li, Ge Li, YunFei Zhao, Jia Li, Zhi Jin, and Hong Mei. Hot or cold? adaptive temperature sampling for code generation with large language models. *arXiv preprint arXiv:2309.02772*, 2023.

A Extended Discussion

A.1 Expanded Related Work

Adaptive decoding through sampling-parameter control. Recent adaptive decoding methods modify stochasticity through temperature, top- p , or uncertainty-derived surrogates. AdaptiveDecoder/LPO (Dhuliawala et al., 2024) predicts a temperature choice from hidden state, while AUTODECO (Wang et al., 2026) predicts continuous token-level temperature and top- p . Heuristic methods such as ADAPT (Zhu et al., 2023), EDT (Zhang et al., 2024), and entropy/confidence-based rules similarly vary sampling parameters using output-distribution statistics rather than a learned controller. These methods are the most direct comparators for our paper because they act on the same low-dimensional post-hoc control surface.

Distributional truncation and target-based sampling. A separate line of work reshapes the sampling distribution using static or locally computed distributional criteria rather than a learned controller. Mirostat (Basu et al., 2021) adapts sampling to hold generated perplexity near a target value; locally typical sampling (Meister et al., 2023) keeps tokens whose information content is close to the conditional entropy; truncation sampling as desmoothing (Hewitt et al., 2022) motivates nucleus-style truncation as removing smoothing mass introduced during training; and min- p sampling (Minh et al., 2025) truncates relative to the top-1 probability rather than at a fixed cumulative mass. These methods share our control surface in that they reshape a frozen next-token distribution, but they set thresholds from properties of the current distribution or from a target statistic, rather than predicting parameters from hidden state. They are therefore closer to the heuristic controllers we evaluate than to AUTODECO, and the same question applies to them: how much of their benefit survives comparison against a fixed configuration matched to their own average induced behavior?

Uncertainty-triggered decoding interventions. AdaDec (He et al., 2025) is closely related in motivation: it identifies high-uncertainty code-generation steps and intervenes selectively. Its mechanism, however, differs from the scalar-control family we study. Rather than predicting a smooth token-level temperature or top- p , AdaDec uses learned entropy thresholds to trigger a pause-then-rerank lookahead procedure at uncertain steps. We therefore view it as adjacent evidence that token-level adaptive inference can help, but not as a directly matched baseline for our decomposition, whose main theory is built around low-dimensional sampling-parameter modulation of a frozen next-token distribution.

Broader adaptive inference-time policies. Other recent work adapts inference-time behavior through control surfaces that are orthogonal to ours. AdaSD (Lu et al., 2025) dynamically adjusts speculative decoding generation length and token-acceptance criteria during inference. This is an adaptive decoding policy, but it targets latency-quality trade-offs in speculative decoding rather than our question of whether token-level sampling-parameter variation outperforms a matched static operating point. Concurrent work on learned adaptive decoding under explicit compute budgets (Appendix A.2) also partly falls outside our setup: its sequence-level policy selects among decoding strategies for an entire prompt, and its token-level policy is trained directly from task reward under budget constraints.

Token-level verification and correction. Token-Guard (Zhu et al., 2026) represents a richer form of token-level control. It performs self-checking during generation, scores hallucination risk, and uses pruning and regeneration to correct suspected errors. This is important context for the scope of our claims. Our results argue for structural limits of *low-dimensional post-hoc sampling control* (temperature/top- p applied to frozen logits), not for the futility of richer token-level mechanisms that can verify, reject, or regenerate candidate content.

Matched baselines and apparent progress. Our result belongs to a recognizable methodological pattern in which a carefully matched baseline accounts for much of the reported advantage of a more complex method. [Melis et al. \(2018\)](#) found that well-tuned LSTM language models matched or exceeded the architectures proposed to replace them; [Lucic et al. \(2018\)](#) found no evidence that any GAN variant dominated the original formulation once the tuning budget was equalized; [Musgrave et al. \(2020\)](#) reported that much of the progress in deep metric learning did not survive a consistent evaluation protocol; and [Ferrari Dacrema et al. \(2019\)](#) found that most neural recommendation results were not reproducible against simple baselines. What these studies share with ours is not the finding but the mechanism: when the baseline is not matched to the proposed method along the dimension that actually carries the gain, the comparison attributes the improvement to the wrong component. MEANSHIFT is our attempt to supply the matched baseline for adaptive decoding.

A.2 Concurrent work

Concurrent with our work, [Su et al. \(2026\)](#) propose *Learning Adaptive LLM Decoding*, a reinforcement-learning framework for adaptive decoding under explicit compute budgets. Their method trains lightweight decoding adapters on a frozen language model using verifiable terminal rewards, with two forms of adaptation: (i) a *sequence-level* adapter that selects among a small set of decoding strategies for an entire prompt, and (ii) a *token-level* adapter that selects among discrete temperature actions during generation.

This paper is complementary to ours rather than directly overlapping. Their primary question is whether learned decoding policies can improve the accuracy–budget trade-off relative to strong static baselines. Our question is different: when adaptive decoding improves over standard fixed decoding, how much of that gain is actually due to token-level control, versus simply discovering a better fixed operating point?

This distinction matters for interpreting their results. Their sequence-level method is best viewed as prompt-level strategy selection among static decoding configurations, which is much closer to operating-point selection than to fine-grained continuous token-level control. Their token-level method is more directly comparable to prior adaptive decoding work, but is evaluated against fixed and mixed static baselines rather than against a matched mean-policy baseline of the kind we advocate here. From our perspective, an important next step would be to ask whether the gains of such learned policies remain after comparison to a fixed decoder that matches the policy’s average induced operating point.

More broadly, [Su et al. \(2026\)](#) strengthen the case that inference-time control can matter, especially when trained directly on task reward and compute constraints. At the same time, our results suggest that the causal interpretation of these gains deserves care. A learned controller may outperform a default static decoder without necessarily demonstrating that its fine-grained residual token-level variation is the main source of improvement. Our inverse-temperature diagnostics provide one possible lens for analyzing this question in future adaptive-decoding systems.

A.3 Adaptive decoding as restricted conditional computation

One possible way to interpret these findings is through the lens of model adaptation. A learned adaptive decoding head is an extremely restricted form of conditional computation: a small module that reads hidden states and modulates the output distribution via exactly two scalar controls. From this perspective, it is unsurprising that the dominant effect is a global shift: we would expect a small adapter over a well-trained model to first absorb the most systematic bias (the operating point) before providing useful instance-level variation. We offer this as a reading of the results rather than a separate finding.

A.4 Adaptive decoding versus adaptation-budget baselines

Our findings raise a further question we flag for future work: when is learned adaptive decoding an efficient use of adaptation budget at all? Training a controller on curated

supervision or reinforcement-learning rewards is itself a form of task adaptation, so once that cost is paid, the relevant comparison is not only against untuned fixed decoding but against other uses of the same data and compute, such as parameter-efficient fine-tuning (Hu et al., 2022). We do not run that comparison here, but our results make it hard to ignore. When a controller’s gains are largely recoverable by a fixed decoder built from its own mean predictions, the head is functioning mainly as an expensive operating-point calibrator. We therefore encourage future adaptive decoding work to distinguish three questions: (i) whether adaptation helps relative to default decoding, (ii) whether it helps beyond a matched MEANSHIFT-style fixed operating point, and (iii) whether it is the best use of the available adaptation budget.

B Proof of Proposition 1: Strict Concavity

We work in inverse temperature $\beta = 1/T$ because β is the natural scalar parameter of the Gibbs/softmax family. For fixed logits z_t , the gold-token teacher-forced log-probability can be written as

$$\ell_t(\beta) = \beta z_{t,y_t} - \log \sum_j e^{\beta z_{t,j}},$$

that is, an affine term minus a log-partition function. The concavity of $\ell_t(\beta)$ is the standard convexity of the log-partition function in its natural parameter (Boyd & Vandenberghe, 2004), specialized to the softmax; the calculation below shows that its second derivative is exactly the negative logit variance under the induced softmax.

Let $q_i(\beta) = e^{\beta z_i} / Z(\beta)$ with $Z(\beta) = \sum_j e^{\beta z_j}$, and write $\bar{z}(\beta) = \sum_i q_i(\beta) z_i$ for the probability-weighted mean logit. From (2),

$$\ell'_t(\beta) = z_{t,y_t} - \bar{z}(\beta). \quad (8)$$

Differentiating again:

$$\ell''_t(\beta) = -\frac{\partial \bar{z}}{\partial \beta} = -\sum_i \frac{\partial q_i}{\partial \beta} z_i. \quad (9)$$

Since $\frac{\partial q_i}{\partial \beta} = q_i(z_i - \bar{z})$, we have

$$\ell''_t(\beta) = -\sum_i q_i(z_i - \bar{z})z_i = -\left(\sum_i q_i z_i^2 - \bar{z}^2\right) = -\text{Var}_{q(\beta)}[z_{t,j}]. \quad (10)$$

Because z_t has at least two distinct entries and $q(\beta)$ has full support ($q_i > 0$ for all i), the variance is strictly positive. Hence $\ell''_t(\beta) < 0$ for all $\beta > 0$. $\square$

C Entropy–Temperature Monotonicity

Proposition 2 (Entropy–temperature monotonicity). *Let $z \in \mathbb{R}^{|V|}$ have at least two distinct entries, and define $H(z, T) = H(\text{softmax}(z/T))$. Then:*

(a) $H(z, T)$ is strictly increasing in T on $(0, \infty)$, with

$$\frac{\partial H}{\partial T} = \frac{1}{T^3} \text{Var}_{p_T}[z_i] > 0, \quad (11)$$

where $p_T = \text{softmax}(z/T)$.

(b) $\lim_{T \rightarrow 0^+} H(z, T) = 0$ and $\lim_{T \rightarrow \infty} H(z, T) = \log |V|$.

Proof. Let $p_i(T) = e^{z_i/T} / Z(T)$ with $Z(T) = \sum_j e^{z_j/T}$, and write $\bar{z}(T) = \sum_i p_i(T) z_i$. Differentiating,

$$\frac{\partial p_i}{\partial T} = p_i(T) \frac{\partial}{\partial T} \left[\frac{z_i}{T} - \log Z(T) \right] = \frac{p_i(T)}{T^2} [\bar{z}(T) - z_i], \quad (12)$$

where we used $\frac{1}{Z} \frac{\partial Z}{\partial T} = -\bar{z}(T)/T^2$.

Now differentiate $H(z, T) = -\sum_i p_i \log p_i$:

$$\frac{\partial H}{\partial T} = -\sum_i \frac{\partial p_i}{\partial T} (\log p_i + 1) = -\sum_i \frac{\partial p_i}{\partial T} \log p_i, \quad (13)$$

where the second equality uses $\sum_i \partial p_i / \partial T = 0$. Substituting $\log p_i = z_i/T - \log Z$ and (12),

$$\frac{\partial H}{\partial T} = -\frac{1}{T^2} \sum_i p_i (\bar{z} - z_i) \left(\frac{z_i}{T} - \log Z \right) \quad (14)$$

$$= -\frac{1}{T^3} \sum_i p_i (\bar{z} - z_i) z_i + \underbrace{\frac{\log Z}{T^2} \sum_i p_i (\bar{z} - z_i)}_{=0} \quad (15)$$

$$= -\frac{1}{T^3} \left[\bar{z}^2 - \sum_i p_i z_i^2 \right] = \frac{1}{T^3} \text{Var}_{p_T}[z_i]. \quad (16)$$

Strict positivity of the variance follows from z having distinct entries and p_T having full support.

For the limits: as $T \rightarrow 0^+$, p_T concentrates on the argmax, so $H \rightarrow 0$. As $T \rightarrow \infty$, $p_T \rightarrow \text{Uniform}(|V|)$, so $H \rightarrow \log |V|$. $\square$

Corollary 1 (Temperature–entropy bijection). *For fixed logits z with at least two distinct entries, the mapping $T \mapsto H(z, T)$ is a bijection from $(0, \infty)$ to $(0, \log |V|)$. Choosing temperature is equivalent to choosing a target entropy level.*

D Additional Experimental Details

D.1 Auxiliary corpus for operating-point estimation and diagnostics

We use held-out samples from the publicly released Dolci-Think-SFT-32B dataset² as an auxiliary corpus for two purposes: (i) estimating the fixed MEANSHIFT operating point from free-generation traces, and (ii) computing teacher-forced diagnostics such as leverage localization and top- p breakpoint statistics.

We choose this corpus because our primary model, DeepSeek-R1-Distill-Qwen-7B, is distilled from the DeepSeek-R1 family, while Dolci-Think contains high-quality long-form reasoning traces generated by a mix of DeepSeek-R1 and DeepSeek-R1-0528. Compared with generic instruction-tuning or web text, this makes Dolci-Think a better-matched auxiliary source for our setting.

We use Dolci-Think only for auxiliary estimation and analysis, not for benchmark evaluation. For MEANSHIFT estimation, we sample prompts and collect free-generation continuations under AUTODECO. For teacher-forced diagnostics, we analyze assistant-side response tokens from held-out traces. Unless otherwise noted, Dolci-Think-based statistics are computed on approximately 1M tokens.

D.2 Model and decoding configuration

For each model we test, AUTODECO head weights are initialized from the publicly released checkpoints³. Base decoding follows each model family’s recommended evaluation or release configuration rather than a single universal setting. Table 6 summarizes the base decoding parameters used for each model, along with the fixed MEANSHIFT operating points estimated from AUTODECO predictions.

²<https://huggingface.co/datasets/allenai/Dolci-Think-SFT-32B>

³<https://huggingface.co/collections/zacks917/autodeco>

Table 6: Base decoding settings and fixed MEANSHIFT operating points by model. Base decoding follows each model family’s recommended release or evaluation configuration. The MEANSHIFT values are the empirical means of AUTODECO’s predicted temperature and top- p over approximately 1M free-generation tokens.

	DeepSeek-R1-Distill-Qwen-7B	Qwen3-30B-A3B-Instruct-2507	GPT-OSS-20B
<i>Base decoding</i>			
Temperature T	0.6	0.7	1.0
top- p	0.95	0.8	1.0
Source	Guo et al. (2025)	Yang et al. (2025)	official repo ⁴
<i>Fixed MEANSHIFT operating point</i>			
$\bar{T}$	0.798	0.861	1.032
$\bar{p}$	0.907	0.843	0.940

MEANSHIFT is estimated from free-generation traces on the auxiliary Dolci-Think corpus described in Appendix D.1, rather than under teacher forcing. Empirically, the resulting mean predicted temperature and top- p are very similar to those obtained under teacher forcing, while free-generation traces are more flexible to collect across model families because they do not require constructing matched teacher-forced corpora for each model. Specifically, we randomly sample prompts from Dolci-Think and run AUTODECO to generate continuations, collecting approximately 1M generated tokens in total. We then compute the empirical mean of AUTODECO’s predicted T and top- p over those generated tokens and use these means as the fixed MEANSHIFT operating point.

D.3 Generalizing MEANSHIFT beyond coordinatewise means

In the main paper, MEANSHIFT fixes temperature and top- p to the empirical coordinatewise means of AUTODECO’s predictions. For the three models we evaluate, this is a sensible approximation because the induced joint (T, p) distributions are relatively regular and do not show strong multimodality (see Figure 3). However, the coordinatewise mean should not be viewed as the most general definition of an “average operating point.”

More generally, let θ_t denote the adaptive decoder parameters at token t , let Θ_{static} denote a chosen family of fixed decoders, and let π_t^{adapt} and $\pi_\theta^{\text{static}}$ denote the corresponding next-token distributions. We can define the best static analogue of an adaptive policy as

$$\theta^* = \arg \min_{\theta \in \Theta_{\text{static}}} \frac{1}{N} \sum_{t=1}^N D(\pi_t^{\text{adapt}}, \pi_\theta^{\text{static}}), \quad (17)$$

where D is a divergence computed on a tractable top- k sketch of the two distributions.

This formulation preserves joint structure: when an adaptive controller uses correlated, multimodal, or higher-dimensional parameterizations, the best static proxy need not coincide with the mean of the raw parameters. It also extends naturally to future adaptive decoders with additional controls beyond temperature and top- p .

One can therefore view the coordinatewise-mean MEANSHIFT baseline used in the main paper as a simple instantiation of a broader projection principle. In settings like ours, the empirical mean is an effective approximation; in richer settings, Equation (17) provides the more faithful comparator.

D.4 Heuristic baseline configurations

Heuristic hyperparameters are tuned on a small held-out development set formed by combining BRUMO25 with a stratified random sample of 30 MMLU-Pro questions excluded

⁴<https://github.com/openai/gpt-oss>

from the reported lite subset, for a total of 60 development problems. This tuning procedure is intentionally favorable to the heuristic baselines because the question domains are similar to the benchmarks we test on. BRUMO25 is used only for development, and the sampled MMLU-Pro questions are disjoint from the reported MMLU-Pro-lite evaluation set. We never tune MEANSHIFT on this set.

For each heuristic family, we evaluate three candidate settings on this development set and compute a simple selection score equal to the unweighted arithmetic mean of dev-set maj@1, maj@8, and pass@8. We use these three metrics because they are the closest development-set analogues of the one-shot, majority-vote, and coverage metrics reported in the main paper. We do not include pass@1 separately because it is identical to maj@1. After selection, the chosen setting is kept fixed across all reported evaluation benchmarks for the primary model.

CONF GATE. CONF GATE is a two-level temperature rule based on model confidence. Let

$$p_{\max,t} = \max_i \text{softmax}(z_t)_i$$

be the maximum probability under the model’s pre-temperature, pre-top- p next-token distribution. CONF GATE switches to greedy decoding when $p_{\max,t}$ exceeds a threshold τ , and otherwise uses a fixed high temperature:

$$T_t = \begin{cases} \text{greedy}, & p_{\max,t} > \tau, \\ T_{\text{high}}, & \text{otherwise.} \end{cases}$$

It does not modify top- p . For DeepSeek-R1-Distill-Qwen-7B, we fix $T_{\text{high}} = 0.9$ and top- $p = 0.907$, sweep $\tau \in \{0.4, 0.5, 0.6\}$ on the development set, and select $\tau = 0.6$.

ENTROPYSHIFT. ENTROPYSHIFT is a linear entropy-based temperature controller around a reference entropy level. Let

$$\hat{H}_t = \frac{H(\text{softmax}(z_t))}{\log |V|}$$

denote the normalized entropy of the model’s pre-temperature, pre-top- p next-token distribution. The controller sets

$$T_t = T_{\text{base}} + \delta (\hat{H}_t - H_{\text{mean}}),$$

and clips the result to

$$[T_{\text{base}} - \delta H_{\text{mean}}, T_{\text{base}} + \delta (1 - H_{\text{mean}})].$$

Thus low-entropy tokens receive a lower temperature than T_{base} , while high-entropy tokens receive a higher one. It does not modify top- p . For DeepSeek-R1-Distill-Qwen-7B, we intentionally anchor ENTROPYSHIFT at the MEANSHIFT operating point, setting $T_{\text{base}} = 0.798$ and top- $p = 0.907$ to the corresponding fixed MEANSHIFT values estimated from the auxiliary corpus. We set $H_{\text{mean}} = 0.072$ to the empirical mean normalized entropy on the same corpus, so that ENTROPYSHIFT applies zero average shift at the reference operating point. This makes the comparison especially clean: relative to MEANSHIFT, the only possible source of improvement is the entropy-conditioned token-level variation itself, rather than a different average decoding configuration. We then sweep $\delta \in \{0.1, 0.2, 0.3\}$ on the development set and select $\delta = 0.2$.

EDT. EDT is an entropy-based dynamic temperature rule that adapts temperature while keeping top- p fixed. Let

$$H_t = H(\text{softmax}(z_t))$$

denote the raw entropy of the model’s pre-temperature, pre-top- p next-token distribution at step t . We implement EDT as

$$T_t = T_0 \cdot \exp\left(\log(N) \frac{\theta}{\max(H_t, 10^{-5})}\right),$$

where θ is the tuned hyperparameter, T_0 is the maximum temperature, and $N \in (0, 1)$ controls the shape of the entropy-to-temperature mapping. For numerical stability, we clamp

H_t from below by 10^{-5} , clip the resulting temperature to $[0, T_0]$, and treat temperatures below 10^{-5} as greedy decoding in the implementation.

For our experiments, we fix $N = 0.8$ following the authors’ recommendations and set $T_0 = 0.798$ and $\text{top-}p = 0.907$ to allow for direct comparison against MEANSHIFT. We sweep $\theta \in \{0.1, 0.2, 0.3\}$ on the development set and choose $\theta = 0.2$.

D.5 Answer extraction and verification

For all evaluated benchmarks, we instruct the model to place its final answer in `\boxed{\}`, and we score the final boxed expression after standard normalization of simple LaTeX variants (e.g., equivalent fraction formatting and minor spacing differences). For multiple-choice benchmarks such as GPQA-Diamond and MMLU-Pro-lite, we take the final answer choice letter and compare it to the gold option label.

In all cases, verification is deterministic and exact. A sample is counted as correct iff the normalized extracted answer matches the reference answer; we do not use a learned verifier or secondary judging model.

D.6 Random seeds and confidence intervals

All main-paper experiments use 8 random seeds unless stated otherwise; for instruction-following benchmarks we use 16 seeds (Appendix E). For each method and benchmark, we first compute the metric separately for each seed run, then report the mean across seeds. Uncertainty in the main result tables is reported as a symmetric 95% confidence interval across seeds using a Student- t interval, i.e.,

$$\bar{x} \pm t_{0.975, n-1} \frac{s}{\sqrt{n}},$$

where n is the number of seeds, $\bar{x}$ is the mean of the per-seed metric values, and s is their sample standard deviation. For $n = 8$, this uses $t_{0.975, 7} = 2.365$.

E Instruction-Following Generalization

We additionally evaluate on IFEval and IFBench to test whether the same qualitative conclusion extends beyond QA-style reasoning and knowledge benchmarks to more open-ended instruction following. Unlike the main benchmarks, these tasks reward adherence to formatting and content constraints rather than only final-answer correctness, so they probe a different regime from the math and knowledge settings emphasized in the main text.

E.1 DeepSeek-R1-Distill-Qwen-7B

For DeepSeek-R1-Distill-Qwen-7B, we evaluate Base, Greedy, AUTODECO, and MEANSHIFT using the official prompt-level and instruction-level strict/loose metrics. Base, AUTODECO, and MEANSHIFT are averaged over 16 random seeds; Greedy is deterministic and is therefore reported once. Table 7 shows that the main qualitative conclusion still holds in this setting: learned token-level control does not outperform strong static decoders.

On IFEval, MEANSHIFT essentially ties Base on the overall average (41.64 for both, to two decimals) and slightly exceeds AUTODECO (41.43). On IFBench, Greedy is strongest on every reported metric, while Base and MEANSHIFT remain nearly identical and both slightly exceed AUTODECO on the average metric. Thus, even in this more open-ended regime, we find no evidence that AUTODECO’s token-level variation yields a meaningful advantage beyond well-chosen fixed decoding configurations. Relative to the main benchmarks, the shift here is not that learned token-level control becomes stronger, but rather that the best static decoder can move toward a more deterministic operating point.

Benchmark	Metric	Base	Greedy	AUTODECO	MEANSHIFT
IFEval	Prompt Strict	33.53±0.48	33.46	33.35±0.31	33.53±0.40
	Prompt Loose	35.82±0.57	35.67	35.70±0.50	35.79±0.41
	Instr Strict	47.43±0.42	47.24	47.20±0.33	47.49±0.40
	Instr Loose	49.76±0.51	49.40	49.49±0.47	49.75±0.46
	Average	41.64±0.47	41.44	41.43±0.36	41.64±0.37
IFBench	Prompt Strict	13.23±0.51	14.67	13.02±0.57	13.23±0.43
	Prompt Loose	16.31±0.60	17.00	16.06±0.49	16.31±0.47
	Instr Strict	14.90±0.56	15.99	14.84±0.61	14.90±0.37
	Instr Loose	19.17±0.58	19.77	19.04±0.51	19.02±0.40
	Average	15.90±0.54	16.86	15.74±0.51	15.87±0.38

Table 7: Instruction-following results for DeepSeek-R1-Distill-Qwen-7B. Base, AUTODECO, and MEANSHIFT report mean $\pm$ 95% CI across 16 seeds; Greedy is deterministic and is therefore shown as a single run. Higher is better.

Benchmark	Metric	Base	Greedy	AUTODECO	MEANSHIFT
IFEval	Prompt Strict	83.68±0.33	82.81	83.46±0.32	83.81±0.44
	Prompt Loose	87.34±0.38	86.14	87.04±0.38	87.38±0.40
	Instr Strict	89.01±0.22	88.49	88.91±0.24	89.18±0.31
	Instr Loose	91.53±0.27	90.89	91.40±0.26	91.66±0.28
	Average	87.89±0.28	87.08	87.70±0.29	88.01±0.34
IFBench	Prompt Strict	31.00±0.55	29.67	30.90±0.70	31.42±0.65
	Prompt Loose	33.90±0.56	33.33	33.83±0.67	34.04±0.57
	Instr Strict	32.56±0.47	30.81	32.63±0.60	33.12±0.59
	Instr Loose	36.26±0.54	35.76	36.54±0.63	36.57±0.56
	Average	33.43±0.50	32.39	33.47±0.61	33.79±0.53

Table 8: Instruction-following results for Qwen3-30B-A3B-Instruct-2507. Base, AUTODECO, and MEANSHIFT report mean $\pm$ 95% CI across 16 seeds; Greedy is deterministic and is therefore shown as a single run. Higher is better.

E.2 Qwen3-30B-A3B-Instruct-2507

For Qwen3-30B-A3B-Instruct-2507, the same qualitative conclusion holds. On IFEval, MEANSHIFT achieves the best mean on all four reported metrics and on the overall average, with Base consistently second. On IFBench, AUTODECO slightly exceeds Base on several rows; however, even when AUTODECO provides a small improvement over the model’s default decoding, that gain is again recovered (and slightly exceeded) by a matched fixed operating point.

The margins here are modest and many confidence intervals overlap, so we interpret the result directionally rather than as evidence of a large instruction-following gap. The important pattern is consistency: across all ten reported rows (IFEval and IFBench, including averages), MEANSHIFT has the highest mean. Unlike DeepSeek-R1-Distill-Qwen-7B on IFBench, Greedy is not competitive here, which suggests that the best static decoder depends on the model family and task regime. What remains stable across both models is the main conclusion: once operating-point correction is controlled for, we still do not see a reliable advantage from AUTODECO’s residual token-level variation on instruction following.

E.3 GPT-OSS-20B

GPT-OSS-20B shows the same core conclusion, but with a somewhat different instruction-following operating regime from the other two models. On IFEval, MEANSHIFT exceeds AUTODECO on all five reported rows, but Greedy is strongest on the two loose metrics and on the overall average. This suggests that, for GPT-OSS-20B in this setting, the best static

Benchmark	Metric	Base	Greedy	AUTODECO	MEANSHIFT
IFEval	Prompt Strict	31.64±0.31	30.68	31.93±0.25	32.15±0.22
	Prompt Loose	38.35±0.70	48.06	43.01±0.84	44.85±0.75
	Instr Strict	45.53±0.26	44.96	45.93±0.20	46.10±0.22
	Instr Loose	51.04±0.57	58.87	55.27±0.63	56.20±0.58
	Average	41.64±0.39	45.64	44.04±0.42	44.83±0.38
IFBench	Prompt Strict	25.58±0.40	26.67	26.56±0.34	26.66±0.31
	Prompt Loose	29.00±0.47	29.67	30.21±0.41	30.55±0.38
	Instr Strict	29.03±0.42	29.94	29.67±0.45	30.15±0.40
	Instr Loose	32.72±0.47	32.85	33.47±0.41	33.80±0.39
	Average	29.08±0.37	29.78	29.98±0.35	30.29±0.34

Table 9: Instruction-following results for GPT-OSS-20B. Base, AUTODECO, and MEANSHIFT report mean $\pm$ 95% CI across 16 seeds; Greedy is deterministic and is therefore shown as a single run. Higher is better.

decoder may lie even closer to a deterministic operating point than to the model’s default stochastic sampling configuration. Even so, the main pattern remains unchanged: once we compare against strong static baselines rather than only Base decoding, the residual token-level variation of AUTODECO does not show a reliable advantage.

On IFBench, MEANSHIFT again exceeds AUTODECO on every reported row and achieves the best mean on four of the five rows, including the overall average; the remaining row (Prompt Strict) is essentially a tie with Greedy. Thus GPT-OSS-20B does not provide a counterexample to our main claim. Relative to Base, learned adaptive control can help, but its gains are still recoverable by a well-chosen static decoder, and in this model family Greedy itself is sometimes the strongest fixed point. The lesson is therefore slightly sharper here than in the other instruction-following tables: the relevant question is not whether AUTODECO beats the default decoder, but whether it beats the best matched static operating point for the model and task. On these results, it does not.

F Pass@k and Maj@k Curves

For conciseness, in our main-paper experiments, we report pass@16 and maj@16. To test whether the main conclusion depends on the specific sample budget used in Table 1, we also plot pass@k and maj@k as functions of the number of sampled completions. MEANSHIFT remains competitive with, and typically slightly above, AUTODECO over the full range of k rather than only at k = 16.

Accuracy-oriented versus coverage-oriented objectives. The decomposition in Section 3.2 bounds local gold-token likelihood, so it speaks most directly to accuracy-oriented metrics such as maj@1 and maj@k, where a single sample must be well calibrated. Coverage-oriented metrics behave differently; since pass@k rewards a diverse set of attempts, token-level variation that lowers expected per-sample likelihood can still raise the probability that at least one sample succeeds. The cross-model summary (Table 3) is consistent with this reading. MEANSHIFT’s macro-averaged advantage is largest on maj@1 and maj@16 (+1.00 and +1.08) and smallest on pass@16 (+0.55), and the one family where AUTODECO retains an aggregate pass@16 edge is Qwen3-30B-A3B-Instruct-2507 (−0.28). This suggests that in these cases, the residual is not buying accuracy, but it may be buying coverage. Objectives that explicitly reward exploration are therefore the most promising place to look for a controller that beats its own projection, since the curvature penalty constrains expected per-sample likelihood and not the spread across samples that pass@k rewards.

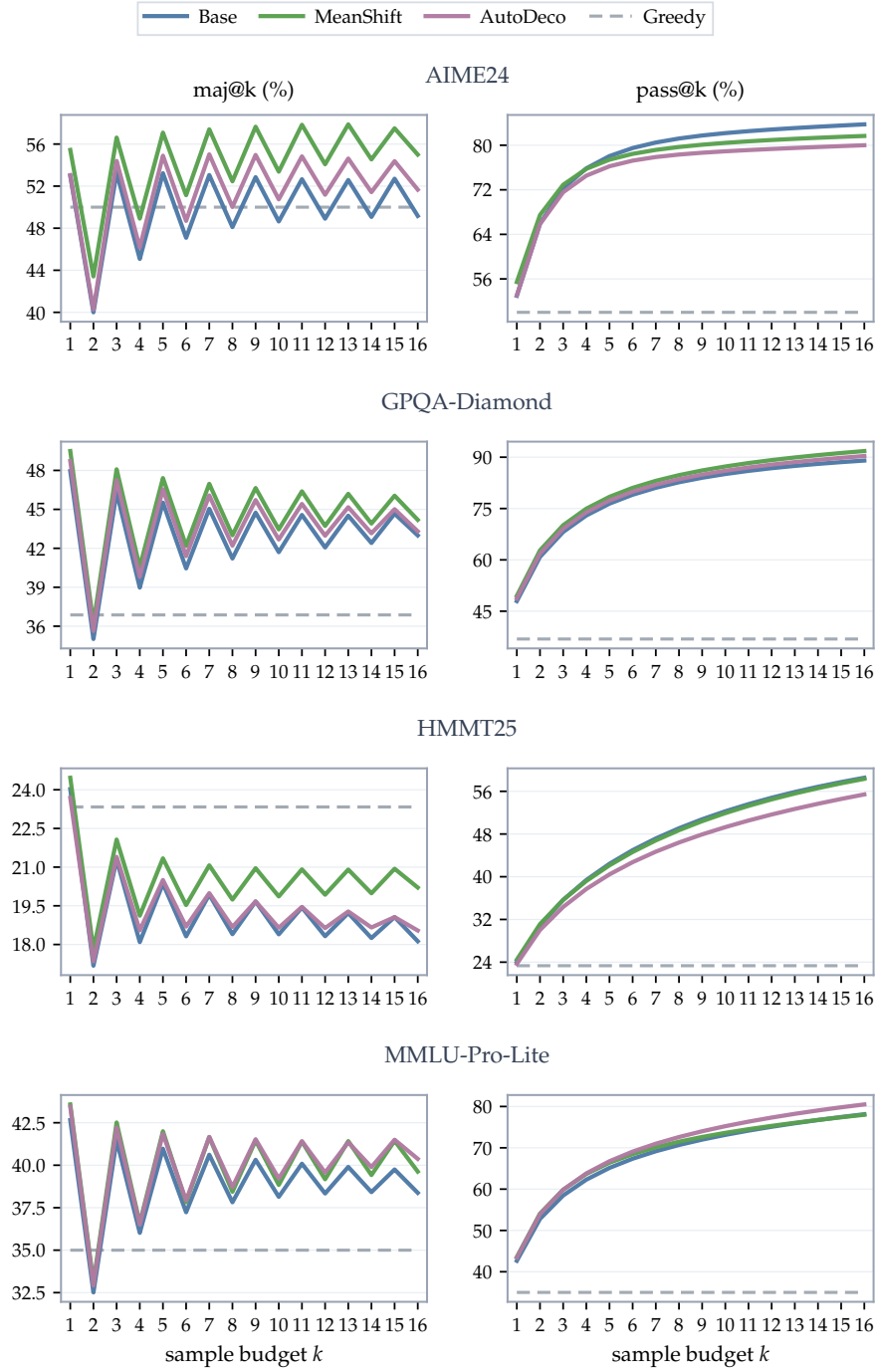


Figure 2: $\text{pass}@k$ and $\text{maj}@k$ curves as the sample budget increases. The overall qualitative pattern matches the main tables: MEANSHIFT tracks or exceeds AUTODECO across the evaluated sample budgets, so operating-point dominance is not an artifact of choosing $k = 16$.

G Inverse-Temperature Alignment: Extended Results

G.1 Formal metric definitions

All diagnostics are computed over held-out tokens where the gold token falls within the top- k logits produced by the model (DeepSeek-R1-Distill-Qwen-7B); we use $k = 128$. For each such token t , write $\beta_t = 1/\hat{T}_t$ for the controller’s predicted inverse temperature and $\bar{\beta}$ for its empirical mean over the training corpus. Let $q_t = \text{softmax}(\bar{\beta} z_t^{(k)})$ denote the softmax over the stored top- k logits at the mean operating point, and let $\bar{z}_t = \sum_i q_t(i) z_{t,i}$ be the probability-weighted mean logit. All quantities below use $\delta_t = \beta_t - \bar{\beta}$, $g_t = z_{t,y_t} - \bar{z}_t$ (the local gradient), and $v_t = \text{Var}_{q_t}[z_{t,i}]$ (the local curvature).

- **Alignment gain** ($\mathbb{E}[g\delta]$): mean of $g_t \delta_t$ across tokens.
- **Curvature penalty** ($\mathbb{E}[v\delta^2]/2$): mean of $\frac{1}{2}v_t \delta_t^2$.
- **Pred. net gain**: $\mathbb{E}[g_t \delta_t - \frac{1}{2}v_t \delta_t^2]$, the second-order Taylor estimate of the per-token gain from Eq. (5).
- **Actual net gain**: $\mathbb{E}[\ell_t(\beta_t) - \ell_t(\bar{\beta})]$, evaluated exactly on the top- k sketch.
- **Corr**(g, δ): Pearson correlation between g_t and δ_t across tokens.
- **Sign agreement**: $\Pr(\text{sign}(\delta_t) = \text{sign}(g_t))$.
- **Cauchy–Schwarz Efficiency**: $\mathbb{E}[g_t \delta_t] / \sqrt{\mathbb{E}[g_t^2] \mathbb{E}[\delta_t^2]}$. Equals 1 iff $\delta_t \propto g_t$; near 0 indicates the residual is nearly orthogonal to the gradient.
- **Corr(pred., actual)**: Pearson correlation between the Taylor-predicted per-token gain and the true gain; validates the second-order approximation.
- **Frac. positive gain**: $\Pr(\ell_t(\beta_t) > \ell_t(\bar{\beta}))$.
- **Penalty > gain**: $\Pr(\frac{1}{2}v_t \delta_t^2 > g_t \delta_t)$.

All expectations are unweighted token-level means over the held-out covered set. Because these quantities use the top- k sketch rather than the full vocabulary, they are exact for the retained support; see the following subsection for coverage statistics.

G.2 Validation of top- k sketch

The alignment diagnostics use a top- k logit sketch to approximate the full softmax. On the evaluation corpus:

- Mean retained probability mass at $T = 1$: 0.979.
- Fraction of rows with retained mass ≥ 0.95 : 90.0%.
- Gold token in top- k rate: 96.8%.

G.3 Teacher-forced diagnostic protocol

All leverage-localization and top- p breakpoint analyses use the auxiliary Dolci-Think corpus introduced in Appendix D.1. For these diagnostics, we randomly sample around 1,000,000 assistant-side tokens and compute all quantities under teacher forcing. We restrict to assistant tokens because the quantities of interest (i.e., entropy, local gradient, curvature, and support changes under adaptive control) are properties of the model’s predictive distribution over response tokens. User-side tokens serve only as conditioning context and are therefore excluded from the tokenwise diagnostics.

These teacher-forced samples are used only for diagnostic analysis, not for MEANSHIFT estimation or for reporting benchmark accuracy. In particular, the leverage-localization statistics in this appendix are computed on the held-out Dolci-Think sample rather than on benchmark test prompts.

G.4 Leverage-localization results and discussion

Entropy bin	Share	Align. gain	Penalty	Net gain	Mean curvature v_t	Support Δ rate
[0.00, 0.05)	59.9%	0.033	0.004	0.029	0.067	0.130
[0.05, 0.10)	15.9%	0.033	0.027	0.006	0.388	0.527
[0.10, 0.20)	15.2%	0.046	0.060	-0.013	0.712	0.769
[0.20, 0.40)	7.6%	0.079	0.154	-0.076	1.224	0.915
[0.40, 0.60)	1.3%	0.065	0.264	-0.199	1.615	0.979
[0.60, 0.80)	0.1%	0.149	0.200	-0.051	1.723	0.991
[0.80, 1.00]	0.0%	N/A	N/A	N/A	N/A	N/A

Table 10: Full leverage localization table by normalized entropy bin on the full teacher-forced evaluation corpus (1,000,612 tokens total; 969,707 covered). Share and Support Δ are computed over all evaluation tokens, while the inverse-temperature leverage terms are averaged over the covered subset within each bin.

Table 10 shows a sharp asymmetric pattern. Most tokens are already easy: 59.9% of tokens fall in the lowest entropy bin and 75.8% fall below normalized entropy 0.10. In this regime, the controller has little useful work to do. The alignment gain is modest (0.033 in both low-entropy bins), the mean curvature is small, and the net gain is near zero by the [0.05, 0.10) bin.

The interesting regime begins once normalized entropy exceeds 0.10. Here the potential leverage is larger, but the curvature penalty rises much faster than the alignment term. The mean curvature increases from 0.712 in the [0.10, 0.20) bin to 1.224 in [0.20, 0.40) and 1.615 in [0.40, 0.60), while the net gain becomes negative in every one of these bins. At the same time, the support-change rate climbs from 0.769 to 0.915 and then 0.979, showing that the controller is most active exactly where the output distribution is most fragile.

In short, adaptive control is cheapest where it matters least, and most valuable where it is hardest to get right.

H Top- p Breakpoint Analysis

Where temperature smoothly rescales the full logit distribution, top- p changes the sampling support only when the cumulative mass crosses a threshold at which one more token enters or leaves the nucleus. We refer to these thresholds as *breakpoints*. Table 11 quantifies this behavior on the teacher-forced diagnostic corpus described in Appendix G.3.

Statistic	Value
Mean $ \Delta p $	0.082
Support change rate	36.9%
$ \Delta N > 1$ rate	23.3%
$ \Delta N > 5$ rate	11.0%
Mean breakpoint dist. (support changed)	0.027
Mean breakpoint dist. (support unchanged)	0.080
Changed / unchanged distance ratio	0.334

Table 11: Top- p breakpoint analysis. Top- p variation is not cosmetic (it changes the support on 36.9% of tokens) but these changes are largely breakpoint-driven.

Table 11 shows that top- p variation is far from inert: support changes occur on 36.9% of tokens, and 11% of tokens see support changes of more than 5 tokens. However, these changes are overwhelmingly driven by proximity to a cumulative-mass breakpoint. Tokens where the support changes are $3\times$ closer to a breakpoint than tokens where it does not (mean distance 0.027 vs. 0.080).

This discrete structure makes smooth optimization of top- p inherently unstable. Small perturbations to p have no effect when the distribution is far from a breakpoint, but can

trigger discontinuous support changes when it is near one. Alongside the curvature penalty on temperature, this helps explain why the learned controller’s modest teacher-forced residual gains do not reliably translate into better free-generation performance.

I Descriptive Analysis of AUTODECO Predictions

We provide descriptive sanity checks for the controller outputs. The goal is not to argue from qualitative plots alone that token-level adaptation is effective, but to rule out a simpler failure mode: that AUTODECO collapses to an almost constant policy or produces obviously pathological parameter values. Figure 3 shows that under free generation, the predicted temperatures and top- p values occupy a broad interior range rather than concentrating at a single point or at clipping boundaries.

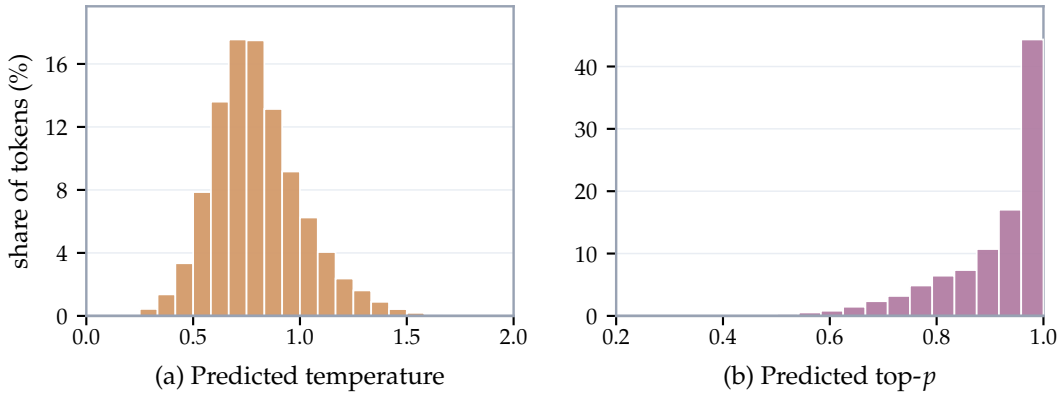


Figure 3: Marginal distributions of AUTODECO predictions over approximately 1M free-generation tokens collected from randomly sampled Dolci-Think-SFT-32B prompts. The controller is non-degenerate and does not simply collapse to a nearly constant operating point.

Figure 4 provides a complementary descriptive view of how AUTODECO responds to simple logit-space statistics under teacher forcing. As expected, the predicted temperature increases with entropy and decreases with the pre-temperature top-1 probability $p_{\max}$, so that the controller tends to inject more randomness when the model is uncertain and sharpen the distribution when it is confident. Figure 5 shows an analogous pattern for top- p : the predicted top- p decreases as entropy rises, which is qualitatively consistent with suppressing diffuse tail mass in high-uncertainty regions, and increases with top-1 confidence, allowing slightly broader sampling when the model already places strong mass on a small set of plausible tokens. Even though these trends are intuitively reasonable, as we show in the main paper, this token-level variation is usually not more useful than a matched, fixed operating point.

J Detailed Cross-Model Generalization Benchmark Results

J.1 Qwen3-30B-A3B-Instruct-2507

Table 12 shows that the operating-point dominance observed in the main model generalizes to larger instruction-tuned models. MEANSHIFT achieves the best mean score on 9 of the 12 benchmark-metric cells, matching or exceeding AUTODECO on the majority of settings. AUTODECO retains a narrow advantage only on pass@16 for AIME24 and GPQA-Diamond, and on maj@16 for MMLU-Pro-lite; in these cases, the confidence intervals often overlap with MEANSHIFT, suggesting the gains are not statistically distinct.

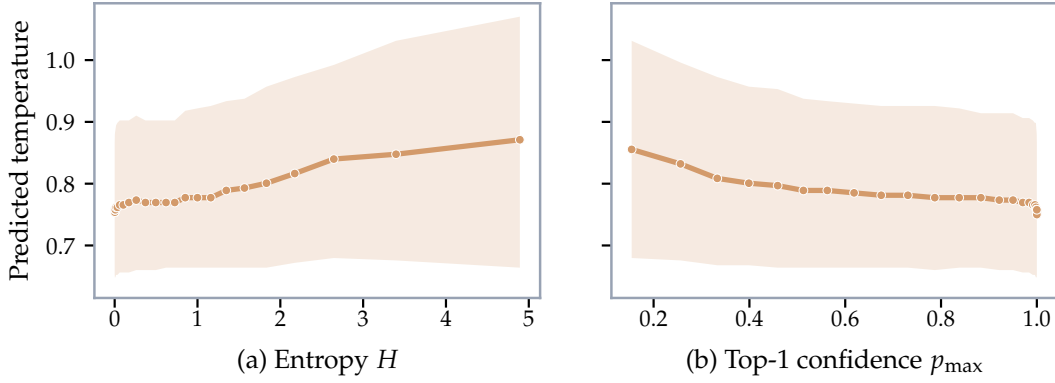


Figure 4: Predicted temperature versus entropy and pre-temperature top-1 probability $p_{\max}$ under teacher forcing. AUTODECO raises temperature on high-entropy, low-confidence tokens and lowers it on low-entropy, high-confidence tokens.

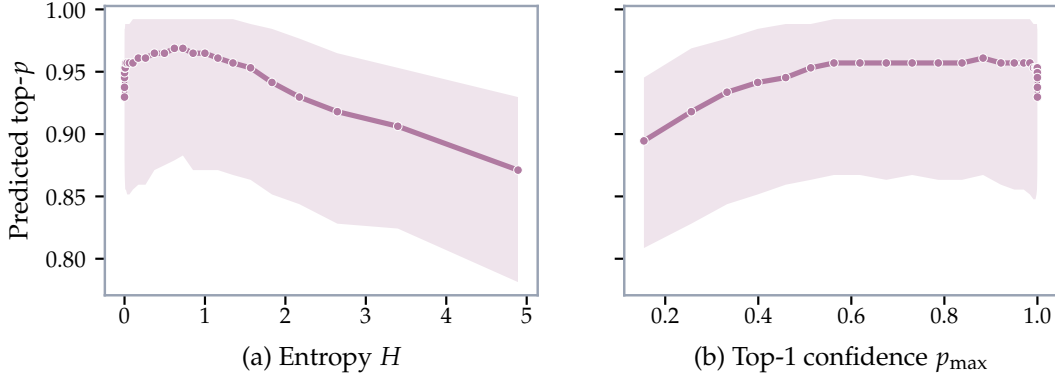


Figure 5: Predicted top- p versus entropy and pre-temperature top-1 probability $p_{\max}$ under teacher forcing. AUTODECO lowers top- p in high-entropy regimes and raises it when the model is more confident.

J.2 GPT-OSS-20B

GPT-OSS-20B supports three levels of “reasoning effort”. For our experiments, we use the medium reasoning effort to match the training setting of the publicly released AUTODECO weights. As Table 13 shows, MEANSHIFT remains competitive with or superior to AUTODECO across almost all metrics. In the few cases where AUTODECO edges out MEANSHIFT, the performance gap is small and frequently within the margin of error. The persistence of this pattern across different model families (DeepSeek, Qwen, GPT-OSS) suggests that the structural barrier to token-level control is not specific to a single model architecture or training recipe, but is inherent to the decoding interface itself.

K Grid Search Operating-Point Recovery

To verify that MEANSHIFT’s operating point does not depend on access to the learned controller, we run a coarse grid search over $T \in \{0.7, 0.8, 0.9\}$ and $p \in \{0.90, 0.95\}$ on the same development set used for heuristic tuning (Appendix D.4), using the same selection score (unweighted mean of dev-set maj@1, maj@8, and pass@8).

Table 15 compares the grid-search-selected configuration against MEANSHIFT and AUTODECO on all four evaluation benchmarks. The grid search configuration matches MEANSHIFT to within 1 pp on every benchmark-metric cell, with the two methods alternating in which is marginally higher. Both consistently exceed AUTODECO (grid search on 10/12 cells,

Benchmark	Metric	Base	Greedy	AUTODECO	MEANSHIFT	Δ
AIME24	maj@1	74.24 $\pm$ 1.13	63.33 $\pm$ 0.00	72.81 $\pm$ 0.86	74.92 $\pm$ 0.87	+2.11
	maj@16	75.83 $\pm$ 1.97	63.33 $\pm$ 0.00	74.17 $\pm$ 2.47	76.25 $\pm$ 1.79	+2.08
	pass@16	89.58 $\pm$ 0.99	63.33 $\pm$ 0.00	89.80 $\pm$ 1.14	89.33 $\pm$ 1.49	-0.67
GPQA-Diamond	maj@1	61.97 $\pm$ 0.26	62.12 $\pm$ 0.00	62.96 $\pm$ 0.44	63.51 $\pm$ 0.31	+0.55
	maj@16	61.49 $\pm$ 0.95	62.12 $\pm$ 0.00	62.69 $\pm$ 0.69	63.13 $\pm$ 1.03	+0.44
	pass@16	89.27 $\pm$ 0.81	62.12 $\pm$ 0.00	89.52 $\pm$ 0.84	88.64 $\pm$ 0.50	-0.88
HMMT25	maj@1	41.71 $\pm$ 0.43	46.67 $\pm$ 0.00	42.27 $\pm$ 0.47	43.03 $\pm$ 0.74	+0.76
	maj@16	41.88 $\pm$ 1.45	46.67 $\pm$ 0.00	44.17 $\pm$ 1.21	44.58 $\pm$ 1.43	+0.41
	pass@16	61.25 $\pm$ 1.43	46.67 $\pm$ 0.00	60.62 $\pm$ 1.97	60.83 $\pm$ 1.89	+0.21
MMLU-Pro-lite	maj@1	75.50 $\pm$ 0.34	75.00 $\pm$ 0.00	76.20 $\pm$ 0.14	76.85 $\pm$ 0.58	+0.65
	maj@16	75.25 $\pm$ 0.87	75.00 $\pm$ 0.00	76.88 $\pm$ 0.70	76.13 $\pm$ 0.82	-0.75
	pass@16	84.50 $\pm$ 0.77	75.00 $\pm$ 0.00	84.00 $\pm$ 0.81	84.24 $\pm$ 0.89	+0.24

Table 12: Detailed per-benchmark results for Qwen3-30B-A3B-Instruct-2507 (mean $\pm$ 95% CI across seeds; higher is better). The final column reports $\Delta = \text{MEANSHIFT} - \text{AUTODECO}$ in percentage points.

Benchmark	Metric	Base	Greedy	AUTODECO	MEANSHIFT	Δ
AIME24	maj@1	75.07 $\pm$ 0.54	80.00 $\pm$ 0.00	75.36 $\pm$ 0.74	77.02 $\pm$ 0.58	+1.66
	maj@16	79.38 $\pm$ 0.97	80.00 $\pm$ 0.00	78.75 $\pm$ 1.28	81.04 $\pm$ 1.25	+2.29
	pass@16	91.67 $\pm$ 0.92	80.00 $\pm$ 0.00	92.92 $\pm$ 0.61	93.12 $\pm$ 0.79	+0.20
GPQA-Diamond	maj@1	66.38 $\pm$ 0.62	60.61 $\pm$ 0.00	66.39 $\pm$ 0.58	67.45 $\pm$ 0.34	+1.06
	maj@16	65.53 $\pm$ 1.19	60.61 $\pm$ 0.00	63.89 $\pm$ 1.46	66.54 $\pm$ 1.19	+2.65
	pass@16	93.69 $\pm$ 1.22	60.61 $\pm$ 0.00	93.18 $\pm$ 0.60	93.50 $\pm$ 0.92	+0.32
HMMT25	maj@1	49.27 $\pm$ 1.01	46.67 $\pm$ 0.00	48.92 $\pm$ 0.78	49.18 $\pm$ 1.07	+0.26
	maj@16	50.62 $\pm$ 1.74	46.67 $\pm$ 0.00	48.54 $\pm$ 1.29	50.42 $\pm$ 1.57	+1.88
	pass@16	81.04 $\pm$ 2.12	46.67 $\pm$ 0.00	81.46 $\pm$ 1.71	81.04 $\pm$ 2.22	-0.42
MMLU-Pro-lite	maj@1	70.52 $\pm$ 0.52	73.00 $\pm$ 0.00	70.09 $\pm$ 0.41	70.87 $\pm$ 0.23	+0.78
	maj@16	68.75 $\pm$ 0.97	73.00 $\pm$ 0.00	70.12 $\pm$ 1.30	69.38 $\pm$ 0.77	-0.74
	pass@16	87.75 $\pm$ 0.74	73.00 $\pm$ 0.00	84.88 $\pm$ 1.22	86.88 $\pm$ 1.04	+2.00

Table 13: Detailed per-benchmark results for GPT-OSS-20B (mean $\pm$ 95% CI across seeds; higher is better). The final column reports $\Delta = \text{MEANSHIFT} - \text{AUTODECO}$ in percentage points.

MEANSHIFT on 10/12 cells), confirming that the controller’s dominant effect (operating-point discovery) is achievable through a trivial sweep. The smoothness of the development-set landscape (Table 14) further suggests a broad “basin” around $T \approx 0.8$, $p \approx 0.9\text{--}0.95$ rather than a sharp optimum requiring learned localization.

L Control-Factorization Ablation on AIME24

To separate the contributions of temperature and top- p , we compare four decoding regimes on AIME24: (i) fully adaptive temperature and top- p (AUTODECO), (ii) adaptive temperature with top- p fixed at the MEANSHIFT operating point, (iii) adaptive top- p with temperature fixed at the MEANSHIFT operating point, and (iv) fully fixed temperature and top- p (MEANSHIFT). Because this ablation is run on AIME24 only, we interpret this only as a mechanism check rather than a standalone generalization claim. For this AIME24-only ablation, we use 16 random seeds rather than 8 to better differentiate each method, since results on AIME24 are a bit noisier across seeds.

Table 16 sharpens the main story. Holding temperature fixed at the MEANSHIFT operating point while retaining adaptive top- p improves over full AUTODECO on all three metrics, whereas retaining adaptive temperature while fixing top- p does not improve the majority

T	p	maj@1	maj@8	pass@8	Selection score
0.7	0.90	50.76±0.88	45.97±1.99	76.94±0.99	57.89
0.7	0.95	51.11±1.20	47.78±2.23	75.28±1.68	58.06
0.8	0.90	52.31±1.98	49.17±3.68	77.08±1.57	59.52
0.8	0.95	53.58±1.50	49.72±2.59	79.72±2.42	61.01
0.9	0.90	52.69±1.46	49.44±2.18	78.06±1.55	60.06
0.9	0.95	51.39±1.24	47.36±1.83	76.39±1.85	58.38

Table 14: Development-set grid search over fixed (T, p) configurations (60 problems, mean $\pm$ 95% CI across seeds). The landscape is smooth, with $(T=0.8, p=0.95)$ achieving the best selection score. This configuration is close to MEANSHIFT’s operating point ($T=0.798, p=0.907$); the learned controller is not needed to identify the right region.

Benchmark	Metric	AUTODECO	MEANSHIFT	Grid search
AIME24	maj@1	53.44±1.46	55.91±1.01	55.70±1.18
	maj@16	52.50±3.25	<u>55.42±1.44</u>	56.67±4.21
	pass@16	78.75±1.44	81.25±1.44	81.25±2.07
GPQA-Diamond	maj@1	48.72±0.35	49.49±0.39	<u>49.38±0.57</u>
	maj@16	43.31±1.35	44.19±0.98	<u>43.75±1.32</u>
	pass@16	<u>90.34±0.48</u>	91.79±1.80	<u>90.34±1.20</u>
HMMT25	maj@1	23.68±0.47	<u>24.47±0.69</u>	25.34±0.56
	maj@16	18.54±1.12	<u>20.21±1.37</u>	20.62±1.33
	pass@16	55.42±2.58	<u>58.33±3.11</u>	59.58±3.03
MMLU-Pro-lite	maj@1	43.45±0.40	43.58±0.49	<u>43.51±0.49</u>
	maj@16	40.38±1.26	39.62±0.77	<u>40.00±0.89</u>
	pass@16	80.50±1.84	79.26±1.55	<u>79.62±1.73</u>

Table 15: Evaluation comparison of grid-search-selected fixed decoding ($T=0.8, p=0.95$) against MEANSHIFT and AUTODECO (mean $\pm$ 95% CI across seeds; higher is better). **Bold** = best mean, underline = second best. Grid search matches MEANSHIFT within 1 pp on every cell and exceeds AUTODECO on 10/12 cells, confirming that the operating-point region is independently discoverable.

metrics. This suggests that, on AIME24, whatever residual adaptive value remains is concentrated more in the top- p head than in the temperature head.

At the same time, neither factorized variant surpasses MEANSHIFT on maj@1 or maj@16, and the best pass@16 is matched by MEANSHIFT. Moreover, full AUTODECO is never the best configuration, indicating no positive interaction between the two residual controls. This is consistent with the breakpoint analysis in Appendix H: any remaining benefit from adaptive top- p appears discrete and limited, while the dominant effect still comes from operating-point correction.

Configuration	maj@1	maj@16	pass@16
Adaptive T , adaptive p (AUTODECO)	52.83 $\pm$ 1.21	51.11 $\pm$ 2.76	79.44 $\pm$ 1.52
Adaptive T , fixed p	52.33 $\pm$ 0.96	50.56 $\pm$ 2.53	81.39 $\pm$ 2.11
Fixed T , adaptive p	53.78 $\pm$ 0.61	53.61 $\pm$ 1.91	81.67$\pm$1.69
Fixed T , fixed p (MEANSHIFT)	55.45$\pm$0.83	55.00$\pm$1.69	81.67$\pm$1.11

Table 16: Control-factorization ablation on AIME24 (mean $\pm$ 95% CI across seeds; higher is better). Fixing temperature at the MEANSHIFT operating point while retaining adaptive top- p recovers most of the remaining adaptive benefit, whereas retaining adaptive temperature while fixing top- p does not improve the majority metrics. MEANSHIFT remains best on maj@1 and maj@16 and ties the best pass@16.