When Is a Single Deployment-Time Pessimism Rule Too Coarse? A Post-Training Audit for Fixed-Backbone Tool-Use Controllers

Anonymous authors
Paper under double-blind review

Abstract

When a tool controller enters change control, its logged trajectories, representation, and value estimator can be frozen while the uncertainty coefficient used at action selection remains configurable. We identify the resolution of that coefficient—one global value or a decision-dependent field—as a post-training control variable. We hold the language model, logged data, state encoder, CQL value network, support statistics, validation choice, and evaluation protocol fixed, and change only this deployment-time rule. On a 15,000-outcome fixed-stack evaluation spanning GSM-8K-Tools, HumanEval-Exec, and NQ-RAG, StrataLCB, a monotone four-signal coefficient field, improves equal-environment mean success from 77.0% to 78.2% and the fixed ensemble-support Q1 success from 73.88% to 77.02% relative to Scalar-LCB. It also lowers the equal-environment unsafe-early-stop mean from 5.5% to 4.0%. A scalar-coefficient mismatch proposition characterizes why heterogeneous ideal penalties cannot in general be represented by a single coefficient. These results establish coefficient resolution as a measurable final intervention for fixed tool controllers operating over heterogeneous state–action support.

1 Introduction

Tool controllers often become only partially editable after deployment. Retraining the language model or value network can trigger a new validation cycle; recollecting trajectories changes the behavior distribution; and changing a tool schema can require an interface review. The rule that converts a fixed uncertainty estimate into an action score can remain configurable after those objects have been frozen. This motivates a focused question: what resolution must that final rule retain when state–action support is heterogeneous?

Adaptive-inference systems commonly learn a router, vary model depth, select among models, or optimize a new policy (Graves, 2016; Elbayad et al., 2020; Xin et al., 2020; Leviathan et al., 2023; Chen et al., 2023; Snell et al., 2024). Tool-agent methods likewise change prompting, search, reward design, or policy training (Schick et al., 2023; Yao et al., 2023; Asai et al., 2024). Offline-RL pessimism usually changes an objective, critic, model, or training-time uncertainty construction (Kumar et al., 2020; Kostrikov et al., 2022; Kidambi et al., 2020; Yu et al., 2020; 2021). These are valuable interventions, but they do not isolate the resolution of a remaining deployment-time uncertainty coefficient after the learned stack is fixed.

We study that isolated intervention. A scalar lower-confidence-bound coefficient is the operational null. StrataLCB instead maps four fixed risk signals—scarce support, disagreement among fixed support estimators, recent tool unreliability, and encoder-space shift—to a monotone coefficient field. The intervention keeps the language model, logs, encoder, CQL critic, bootstrap heads, support object, tuning choice, and evaluation population unchanged. Its outcome therefore directly tests whether a global coefficient under-resolves heterogeneous decisions.

The paper contributes:

(i) an intervention contract that makes coefficient resolution identifiable after the learned controller stack has frozen;

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.

Table 1: Frozen-stack intervention contract. All listed learned objects and evaluation choices are held fixed between Scalar-LCB and StrataLCB.

Object	*Fixed quantity or changed rule*
Learned stack	FP8 Llama-3.1-70B-Instruct, logged trajectories, state encoder, 0.41M CQL network, five bootstrap heads, training recipe, and tuning choice
Support object	Four fitted support estimators and their cached median and dispersion features
Evaluation	Three in-house environments, five seeds, 1,000 held-out items per environment and seed
Deployment rule	Scalar-LCB uses one coefficient; StrataLCB changes only the coefficient to the field in Eq. 2

(ii) a scalar-coefficient mismatch result and a constrained four-signal coefficient field; and

(iii) a fixed-stack evaluation with explicit population, aggregation, and safety definitions, showing a concentrated weak-support improvement together with lower unsafe early stopping.

2 Problem and Intervention Contract

We model action selection as a budgeted POMDP (Altman, 1999; Carrara et al., 2019; Le et al., 2019) with tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, C, B)$ and common action space

$$\mathcal{A} = \{\texttt{continue}, \texttt{retrieve}, \texttt{verify}, \texttt{stop}\}.$$

Terminal reward is binary success and intermediate rewards are zero. The budget mask applies action costs separately from the learned success value. In HumanEval-Exec, `verify` invokes sandboxed execution; in the retrieval settings it invokes an answer–evidence consistency check.

The fixed epistemic state is a two-layer MLP feature map ($4096 \to 256 \to 64$, GELU) of the last 16 decoder states, four recent tool outcomes, remaining budget, entropy of a shared eight-completion bundle, and step count. Four support estimators are fitted once: $K{=}256$ encoder clusters, a 512-cell VQ codebook, a 100-tree kd-forest, and a tool-type by state-bin factorization. Estimator i supplies

$$r_n^{(i)}(s,a) = \min\left(1, \frac{c_0}{n^{(i)}(s,a)+1}\right),$$

where larger values indicate weaker coverage. The policy receives the median $\bar{r}_n$ and median absolute deviation disp_n from this fixed support object. The VQ, kd-forest, and factorized statistics are precomputed in a cluster–action cache; inference reads the cached features without updating the support estimator.

Both compared rules use the same 0.41M CQL network: a three-layer MLP with hidden widths $[256, 128]$ and five bootstrap heads. It is trained with the common objective

$$\mathcal{L}_{\mathrm{CQL}} = \alpha\, \mathbb{E}_s\left[\log \sum_a e^{\hat{Q}(s,a)} - \mathbb{E}_{a\sim\pi_\beta}\hat{Q}(s,a)\right] + \frac{1}{2}\mathbb{E}_{\mathcal{D}}\left[\hat{Q} - r - \gamma \max_{a'} \hat{Q}'(s',a')\right]^2, \quad (1)$$

with $\alpha = 0.5$, AdamW learning rate $5{\times}10^{-4}$, weight decay 10^{-4}, batch size 1024, 30K steps, and soft-target coefficient $5{\times}10^{-3}$. The values and support object are trained or fitted before the deployment-rule comparison.

3 Localized Pessimism

Let $\mu(s,a)$ and $\sigma(s,a) > 0$ be the mean and standard deviation of the fixed bootstrap ensemble. An ideally calibrated rule uses $h^\star(s,a)\sigma(s,a)$, where $h^\star$ is the coefficient field needed to equalize tail risk across decisions. Its heterogeneity is $\kappa_h = \sup h^\star / \inf h^\star$.

Proposition 1 (Scalar coefficient mismatch)**.** *For any scalar $h > 0$ and coefficient field satisfying $0 < h^\star_{\min} \le h^\star(s,a) \le h^\star_{\max} < \infty$, the worst multiplicative mismatch obeys*

$$\max\left\{\sup_{s,a} \frac{h}{h^\star(s,a)}, \sup_{s,a} \frac{h^\star(s,a)}{h}\right\} \ge \sqrt{\kappa_h}.$$

Table 2: Primary fixed-stack comparison. Mean success and unsafe early stop are equal-environment averages over the 15,000-outcome population. Ensemble-Q1 uses the fixed support object.

Rule	*Mean success*	*Ens.-Q1*	*Unsafe stop*
Scalar-LCB	77.0%	73.88%	5.5%
StrataLCB	78.2%	77.02%	4.0%
StrataLCB − Scalar-LCB	+1.2 pp	+3.14 pp	−1.5 pp

Equality is attained at $h = \sqrt{h^\star_{\min} h^\star_{\max}}$.

The proposition isolates the functional limitation of a scalar coefficient: it applies the same ensemble uncertainty $\sigma(s, a)$ in both cases and varies only its multiplier. Empirical-Bernstein concentration motivates scarcity and dispersion signals, while recent tool reliability and encoder shift add separate fixed deployment signals (Maurer & Pontil, 2009).

StrataLCB is the constrained coefficient field

$$\begin{aligned}\beta_{\mathrm{StrataLCB}}(s,a) = \beta_0(1+\lambda_n \bar{r}_n(s,a))(1+\lambda_d \mathrm{disp}_n(s,a)) \\ \cdot (1+\lambda_r(1-\rho(a)))(1+\lambda_{\mathrm{ood}}\omega(s)),\end{aligned} \tag{2}$$

where $\rho(a)$ is the fixed recent reliability score and $\omega(s)$ is the fixed encoder-shift score. We use $\beta_0 = 0.35$, $c_0 = 16$, and $(\lambda_n, \lambda_d, \lambda_r, \lambda_{\mathrm{ood}}) = (1.5, 0.6, 0.7, 0.9)$, selected once on the GSM-8K-Tools validation seed. Scalar-LCB uses the same action score with $\beta_{\mathrm{LCB}} = 0.5$:

$$a^\star_t = \arg \max_{a:\, \sum_{\tau \le t} C(a_\tau) \le B} \left[\mu(s_t, a) - \beta(s_t, a)\sigma(s_t, a)\right]. \tag{3}$$

Thus the comparison changes the coefficient β, not the value model, action space, logged data, or budget mask.

4 Experimental Protocol

The evaluation uses GSM-8K-Tools, HumanEval-Exec, and NQ-RAG. GSM-8K-Tools combines GSM8K (Cobbe et al., 2021) with a theorem retriever and symbolic verifier; HumanEval-Exec uses sandboxed Python execution (Chen et al., 2021); NQ-RAG uses Contriever over Wikipedia passages (Kwiatkowski et al., 2019; Izacard et al., 2022). Each environment has 18K logged trajectories generated with action mixture $(0.40, 0.25, 0.25, 0.10)$. The base model is FP8 Llama-3.1-70B-Instruct (Dubey et al., 2024). Serving uses four NVIDIA H100 80GB SXM5 devices with CUDA 12.4, PyTorch 2.3.1, NCCL 2.20, and vLLM 0.5.2 (Kwon et al., 2023).

The primary population contains five evaluation seeds $\{11, 23, 47, 89, 137\}$, three environments, and 1,000 held-out items per environment and seed, for 15,000 evaluated outcomes. Mean success and unsafe early-stop rates are first averaged within environment across seeds and items, then weighted equally across the three environments. Ensemble-support Q1 is the lowest quartile under the same fixed four-estimator support object. Terminal success is $r_T = 1$. Unsafe early stop is

$$\mathbb{1}\left[a_T = \texttt{stop} \wedge r_T = 0 \wedge \frac{\sum_{t=1}^{T} C(a_t)}{B} < 0.5\right].$$

We report five-seed averages; the table retains the population and aggregation rule needed to interpret every displayed value.

5 Results

Table 2 is the central fixed-stack result. StrataLCB increases success by 1.2 percentage points under equal environment weighting and has its largest displayed separation in the lowest fixed-support quartile (+3.14 percentage points). Unsafe early stopping also falls under the same intervention. Because all learned objects and the evaluation population are held fixed, this comparison identifies the effect of coefficient resolution rather than of a stronger controller or a changed data source.

Table 3: Environment-level trace for the equal-environment aggregates in Table 2. Success and unsafe-stop entries are five-seed environment means; the final row is their unweighted average.

Environment	*Scalar success*	*Strata success*	*Scalar unsafe*	*Strata unsafe*
GSM-8K-Tools	80.3%	81.6%	5.1%	3.7%
HumanEval-Exec	74.6%	75.9%	7.2%	5.2%
NQ-RAG	76.1%	77.2%	4.3%	3.1%
Equal-environment mean	77.0%	78.2%	5.5%	4.0%

Table 3 makes the aggregate traceable to its environment-level components. The equal-weight success mean is $(81.6 + 75.9 + 77.2)/3 = 78.23\%$, reported as 78.2% in Table 2; the corresponding Scalar-LCB mean is $(80.3 + 74.6 + 76.1)/3 = 77.00\%$. Unsafe early-stop means are computed the same way from the displayed environment values. This explicit aggregation prevents a pooled outcome count, a paired ledger, or a different support slice from being interpreted as the primary mean effect.

6 Discussion

The intervention is useful precisely when support is heterogeneous: Eq. 1 shows that a single multiplier must mismatch at least one region whenever the ideal coefficient field varies, and the fixed-support-Q1 result exposes that separation empirically. The controlled comparison does not change the learned controller or its logged data, so it provides a direct operational diagnostic for deciding whether a scalar deployment rule has too little resolution for its fixed support geometry.

7 Conclusion

Coefficient resolution is a distinct post-training design variable for frozen tool controllers. In a fixed CQL stack, a monotone coefficient field improves equal-environment mean success and fixed-support-Q1 success while reducing unsafe early stopping. The result supplies an identifiable final intervention when change control fixes the learned objects but leaves the action-selection rule configurable.

Reproducibility Statement

Sections 2–4 specify the controller, objective, state features, support object, frozen-versus-changed intervention contract, environments, action space, logging mixture, seeds, population, aggregation, and metrics. Appendix A.1 gives the full proof of the coefficient mismatch result.

AI Use Statement

Generative AI was used solely for manuscript writing, editing, and presentation; the authors reviewed the resulting edits and remain responsible for the final content.

References

Eitan Altman. *Constrained Markov Decision Processes*. CRC Press, 1999. URL https://www.routledge.com/Constrained-Markov-Decision-Processes/Altman/p/book/9781315140223.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.

Nicolas Carrara, Edouard Leurent, Romain Laroche, Tanguy Urvoy, Odalric-Ambrym Maillard, and Olivier Pietquin. Budgeted reinforcement learning in continuous

state space. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.nips.cc/paper_files/paper/2019/hash/4fe5149039b52765bde64beb9f674940-Abstract.html.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023. URL https://arxiv.org/abs/2305.05176.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783v1*, 2024. URL https://arxiv.org/abs/2407.21783v1.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual_2020/poster_SJg7KhVKPH.html.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. URL https://arxiv.org/abs/1603.08983.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. URL https://arxiv.org/abs/2112.09118.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21810–21823, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f7efa4f864ae9b88d43527f4b14f750f-Abstract.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/5941.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3703–3712, 2019. URL https://proceedings.mlr.press/v97/le19a.html.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. In *Proceedings of the 22nd Conference on Learning Theory (COLT)*, 2009. URL https://www.cs.mcgill.ca/~colt2009/papers/012.pdf.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2246–2251, 2020. doi: 10.18653/v1/2020.acl-main.204. URL https://aclanthology.org/2020.acl-main.204/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/11003.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: Conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28954–28967, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f29a179746902e331572c483c45e5086-Abstract.html.

A.1 Proof of the Scalar-Coefficient Mismatch

Proof of Proposition 1. Let $h^\star_{\min} = \inf h^\star$ and $h^\star_{\max} = \sup h^\star$. For a scalar h, the mismatch is at least

$$\max\left\{\frac{h}{h^\star_{\min}}, \frac{h^\star_{\max}}{h}\right\}.$$

Their product is $h^\star_{\max}/h^\star_{\min} = \kappa_h$, so their maximum is at least $\sqrt{\kappa_h}$. Equality holds exactly when the two terms are equal, which gives $h = \sqrt{h^\star_{\min} h^\star_{\max}}$. □