BARM: BAYESIAN ASSISTANCE ROUTING FOR EMBODIED ROBOT MANIPULATION OVER VLM AND HUMAN EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied agents deployed in open-world environments must decide, for each incoming task, whether to execute it autonomously with a lightweight policy, invoke a vision-language model planner, or request human assistance. These execution modes differ in capability and cost, and the routing controller is typically trained on atomic tasks but deployed on compositional, multi-step instructions that lie outside its training distribution. We cast embodied assistance routing as a contextual bandit problem with heterogeneous arm costs and identify two desiderata for sample-efficient routing under distribution shift: distance-aware uncertainty, so that unseen tasks are recognized as uncertain, and probability-matching exploration, so that uncertainty translates into cost-aware action. We propose BARM (Bayesian Assistance Routing for embodied robot Manipulation), a unified Bayesian framework whose single belief state simultaneously satisfies both desiderata: its epistemic variance grows with distance from the training distribution, and Thompson Sampling on this belief realizes probability-matching exploration over net rewards. Reward modeling, exploration, and online adaptation all emerge from this one belief rather than from separately engineered heuristics. We also establish a sublinear Bayesian regret bound for BARM under the well-specified regime (linear reward). On CALVIN and LIBERO benchmarks, BARM achieves the highest cumulative reward against 22 baselines spanning failure detection, reinforcement learning, and contextual bandit methods.

1 INTRODUCTION

How should a language-instructed robot (Ahn et al., 2022; Liang et al., 2023) decide, for each new task, whether to execute it autonomously (Chi et al., 2025; Reuss et al., 2024), consult a more capable planner, or request human assistance (Kelly et al., 2019; Ahn et al., 2024; Ren et al., 2023)? Making this choice reliably across tasks is difficult for embodied AI systems deployed in open-world environments (Mees et al., 2022; Liu et al., 2023), where execution modes with varying capabilities and costs must be orchestrated online. We cast this as a decision-making problem within the embodied stack (Siciliano et al., 2008; Kaelbling & Lozano-Pérez, 2011): the underlying execution modes, whether learned policies, vision-language model (VLM) planners, or human operators, are treated as 'black-box', and the routing controller determines which to invoke for each task of interest.

Such routing decisions are increasingly prevalent in modern embodied stacks, which integrate execution modes with markedly different capability profiles. Lightweight diffusion policies (Chi et al., 2025; Reuss et al., 2024) act in milliseconds on familiar atomic tasks; VLM planners (Ahn et al., 2022; Liang et al., 2023) decompose complex instructions that exceed the capacity of reactive policies; human operators (Kelly et al., 2019; Ahn et al., 2024) can resolve almost any ambiguity when clarification is needed. Each mode is effective within its intended regime, but the routing controller cannot identify that regime in advance. This challenge is particularly relevant when the underlying executors are fixed or costly to retrain, while deployment provides only sparse episode-level outcome feedback. We instantiate this setting with a controlled atomic-to-compositional shift, where the router is initialized from atomic-task outcome data and then encounters unseen compositional instructions. The router must use deployment feedback to learn the cost-adjusted value of each mode and selectively invoke more capable assistance when its additional value justifies the cost.

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.

Prior work addresses parts of this problem. Detection methods (Ren et al., 2023; Römer et al., 2025; Yue et al., 2024; Li et al., 2025) flag when autonomous execution is likely to fail, but their thresholds are calibrated offline and do not indicate which mode is best for the cost. Reinforcement learning (RL) based approaches (Schulman et al., 2017; Mnih et al., 2015; Zhou et al., 2025) optimize routing for reward directly, yet sparse signals and the absence of a per-mode reward model make adaptation slow in the few-hundred-episode regime typical of robotic deployment. Contextual bandit methods (Li et al., 2010; Zhang et al., 2020; Ban et al., 2021) model per-arm reward with online updates and fit the low-data regime well, but their posteriors inherit the geometry of the underlying features: when the feature map collapses semantically distinct inputs to nearby points (Liu et al., 2020), the resulting uncertainty is miscalibrated on out-of-distribution (OOD) contexts and misdirects the exploration policy itself, not merely a confidence estimate.

We identify two desiderata for any Bayesian routing framework operating under this distribution shift. The first is *distance-aware uncertainty*, meaning that the router's epistemic variance on a new context grows with its distance from the training distribution, so that unseen compositional tasks are recognized as uncertain. The second is *calibrated probability-matching exploration*, meaning that arm selection is drawn in proportion to the posterior belief over which arm is cost-optimal, so that uncertainty translates into cost-aware action. We refer to the combination of these two properties as *calibrated Bayesian routing*. The two are complementary: features without an exploration mechanism cannot act on the uncertainty they encode, and exploration without distance-aware features misallocates effort on OOD inputs. A comprehensive review of related work, covering failure detection and help-seeking, learned routing and reinforcement learning, contextual bandits, and distance-aware uncertainty estimation, is provided in Appendix A.

Contributions. Motivated by these desiderata, we introduce **BARM** (**B**ayesian **A**ssistance **R**outing for embodied robot **M**anipulation), a framework that operates strictly at the decision-making layer. BARM satisfies both desiderata via a single Bayesian belief: reward modeling, exploration, and online adaptation all derive from one posterior, in line with the Bayesian agentic control perspective of (Papamarkou et al., 2026). Figure 5 in the supplementary file conceptually contrasts representative routing paradigms and highlights the structural components that distinguish BARM.

The contributions of this work are summarized as follows:

- We formalize embodied assistance routing as a decision-making-layer problem with heterogeneous-cost black-box execution modes under an atomic-to-compositional distribution shift, and identify the two desiderata above as jointly necessary for sample-efficient routing.
- We propose BARM, which realizes both desiderata in a single Bayesian framework: a distance-aware feature backbone, multi-head Variational Bayesian Last-Layer (VBLL) posteriors (Harrison et al., 2024) trained offline via the evidence lower bound (ELBO) and updated online by closed-form conjugate updates, and Thompson Sampling on the resulting posterior for cost-aware arm selection.
- We provide a Bayesian regret analysis for BARM and show that its cumulative regret is sublinear in the online horizon under the well-specified linear reward model on the distance-aware features.
- We validate BARM on CALVIN and LIBERO against 22 baselines spanning detection, RL, and contextual bandits, achieving the highest cost-adjusted reward in all settings; ablations show that removing either desideratum significantly degrades performance.

Notation. Scalars are denoted by lowercase letters, vectors by boldface lowercase, and matrices by boldface uppercase. We write $\mathcal{N}(\cdot\,;\,\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, $\mathrm{KL}(\cdot\|\cdot)$ for the Kullback–Leibler divergence, and $\mathrm{Tr}(\cdot)$ for the matrix trace. Throughout, the sample index is $n \in \{1, \ldots, N\}$, the arm (expert) index is $k \in \{1, \ldots, K\}$, and the episode index is $t \in \{1, \ldots, T\}$.

2 Problem Formulation

Consider an embodied agent that, upon receiving a new task, must choose between K execution modes with different capabilities and costs. In this work, the agent observes a task context consisting

of a scene image s_t and a language instruction ℓ_t at each episode t, and selects one of three modes: (1) executing a diffusion policy directly (fast, no additional cost); (2) invoking VLM-assisted subtask decomposition (moderate cost $c_2 > 0$); or (3) requesting human demonstration (highest cost $c_3 > c_2$). After execution, the agent observes a stochastic reward $r_t \in [0, 1]$ reflecting task completion, where the stochasticity arises because each execution mode is itself a stochastic mapping: the diffusion policy samples actions from a learned distribution, the VLM planner produces variable subtask decompositions, and contact dynamics add further randomness. The central challenge is that the best mode depends on both the task and the capability of each backend, and this mapping is unknown a priori: selecting an overly simple mode for a complex task leads to failure, while unnecessarily escalating to an expensive mode wastes resources.

Here, we adopt a practical deployment setting in embodied robot manipulation tasks: the routing controller is trained *offline* on atomic (single-step) manipulation tasks and evaluated *online* on a mixture of seen atomic tasks and unseen compositional (multi-step) tasks (Mees et al., 2022). This induces a structural, not incidental, distribution shift between training and deployment: the combinatorial space of multi-step instructions cannot be covered by offline data collected on atomic primitives. Let $\mathcal{D}_{\text{simple}} = \{(\mathbf{x}_n, a_n, r_n)\}_{n=1}^{N}$ denote the offline training set collected from N atomic task episodes, where $\mathbf{x}_n \in \mathbb{R}^{D_0}$ is a context feature described in Section 3.1, $a_n \in \{1, \ldots, K\}$ is the execution mode used to collect sample n during offline data collection, and $r_n \in [0, 1]$ is the corresponding observed reward. During online evaluation, episodes are sampled from a mixture of atomic and compositional tasks. Through interaction, the controller must realize that compositional tasks, absent from $\mathcal{D}_{\text{simple}}$, benefit from more capable, albeit costlier execution modes. This cold-start problem motivates the need for both calibrated uncertainty (to recognize that compositional tasks are out of distribution) and principled exploration (to learn the reward structure of each arm/execution mode on these novel task types).

To solve this online routing problem, we formalize this as a contextual bandit with K arms corresponding to the K execution modes and heterogeneous per-use costs $\{c_k\}_{k=1}^{K}$. At each episode t, the agent selects arm $a_t \in \{1, \ldots, K\}$ and receives net reward $r_t - c_{a_t}$. The objective is to maximize $\sum_{t=1}^{T}(r_t - c_{a_t})$ over T episodes. We denote the conditional expected reward of arm k by $\bar{r}^k(s_t, \ell_t) = \mathbb{E}[r_t \mid a_t = k, s_t, \ell_t]$, which must be learned from interaction. The routing problem reduces to learning the context-to-arm mapping that balances the cost of exploration against the cost of suboptimal arm selection. Figure 1 illustrates the three task regimes considered in our embodied assistance routing setting. **Note.** The current formulation operates at the instruction level, with each mode executing its own long-horizon closed-loop trajectory.

3 METHOD

BARM combines a VLM-based feature extractor, a distance-aware backbone, a multi-head VBLL reward model, and a Thompson sampling (TS) routing rule with online posterior updates. The feature extractor produces an instruction-aware context feature for each task (Section 3.1), and the backbone (Liu et al., 2020) maps this feature to a distance-aware representation (Section B). On top of this representation, we maintain a VBLL posterior over the reward of each arm, trained offline by ELBO maximization (Harrison et al., 2024) (Section 3.2). At deployment, an arm is selected by TS on the current posterior, and the posterior of the selected arm is refreshed by a closed-form conjugate update after the reward is observed (Sections 3.3–3.4). Figure 2 summarizes the offline training stage and the online routing loop.

3.1 MULTI-HEAD REWARD MODEL WITH INSTRUCTION- AND DISTANCE-AWARE FEATURE EXTRACTION

The task context (s_n, ℓ_n) at each episode is high-dimensional and multimodal. We extract a compact representation using a frozen VLM encoder via

$$\mathbf{x}_n = \text{VLM}(s_n, \ell_n) \in \mathbb{R}^{D_0}, \tag{1}$$

where D_0 is the encoder output dimension. The raw VLM feature $\mathbf{x}_n$, however, is not distance-aware: its representation was optimized for visual-language recognition rather than uncertainty calibration, and standard backbones can compress semantically dissimilar inputs to nearby points in feature

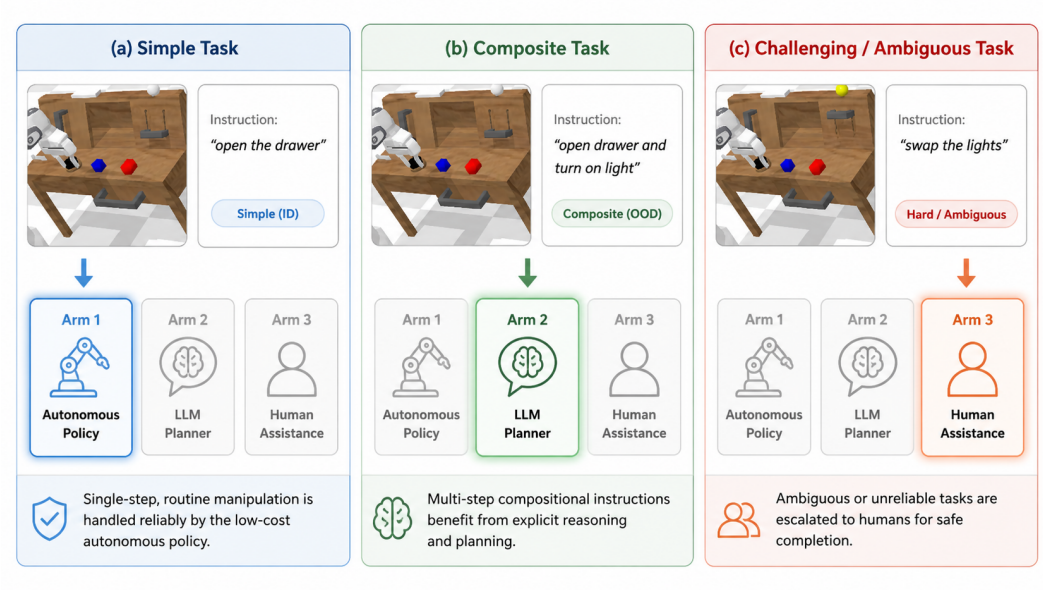


Figure 1: Illustration of the three task regimes in embodied assistance routing. Simple in-distribution tasks are typically handled by the low-cost autonomous policy (Arm 1), compositional out-of-distribution tasks benefit from explicit planning (Arm 2), and challenging or ambiguous tasks are escalated to human assistance (Arm 3).

space (Liu et al., 2020). To alleviate this, we rely on the so-termed spectral-normalized Gaussian process (SNGP) base backbone (described in Section B in the supplementary file) to map $\mathbf{x}_t$ to a distance-aware representation

$$\boldsymbol{\phi}_n := \boldsymbol{\phi}(\mathbf{h}(\mathbf{x}_n; \boldsymbol{\theta}_h); \boldsymbol{\theta}_\phi) \in \mathbb{R}^{D_L}, \tag{2}$$

where $h : \mathbb{R}^{D_0} \to \mathbb{R}^{D_{L-1}}$ is the spectrally-normalized multi-layer perceptron (MLP) with trainable parameters $\boldsymbol{\theta}_h$ (D_{L-1} the penultimate-layer width) and $\boldsymbol{\phi}(\cdot) : \mathbb{R}^{D_{L-1}} \to \mathbb{R}^{D_L}$ is the so-termed random fourier feature (RFF) layer with trainable $\boldsymbol{\theta}_\phi$. Here, $\boldsymbol{\phi}_n$ is shared across all K reward heads. Then we rely on the Bayesian Last-Layer (BLL) regression model (Harrison et al., 2024) to model the reward per arm as

$$r_n^k = \boldsymbol{\beta}^{k\top} \boldsymbol{\phi}_n + \eta_n^k, \quad \eta_n^k \sim \mathcal{N}(0, \sigma_{\eta,k}^2), \tag{3}$$

where $\boldsymbol{\beta}^k \in \mathbb{R}^{D_L}$ is the last-layer weight vector with Gaussian prior $p(\boldsymbol{\beta}^k) = \mathcal{N}(\mathbf{0},\, \sigma_\beta^2 \mathbf{I}_{D_L})$, and $\sigma_{\eta,k}^2$ is a learnable noise variance.

3.2 Offline Training of the Reward Model

Building on equation 3, our next step is to jointly learn the posterior of $\boldsymbol{\beta}^k$ along with other hyperparameters given the aforementioned offline dataset $\mathcal{D}_{\text{simple}} = \{(\mathbf{x}_n, a_n, r_n)\}_{n=1}^N$. To effect scalability, we place a variational Gaussian posterior $q(\boldsymbol{\beta}^k) = \mathcal{N}(\boldsymbol{\mu}^k,\, \boldsymbol{\Sigma}^k)$ over the last-layer weights, which is trained by maximizing the evidence lower bound (ELBO)

$$\mathcal{L}^{\text{ELBO}}(\boldsymbol{\Theta}_k) = \mathbb{E}_{q(\boldsymbol{\beta}^k)} \left[\sum_{n \in \mathcal{S}_k} \log p(r_n^k \mid \boldsymbol{\beta}^k, \boldsymbol{\phi}_n) \right] - \mathrm{KL}\big(q(\boldsymbol{\beta}^k) \| p(\boldsymbol{\beta}^k)\big)\,, \tag{4}$$

where $\boldsymbol{\Theta}_k := \{\boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k, \sigma_{\eta,k}^2\}$ and $\mathcal{S}_k := \{n : a_n = k,\ n \in \{1, \ldots, N\}\}$ collects the training sample indices assigned to arm k. Under Gaussian conjugacy, this ELBO admits a closed-form expression (Harrison et al., 2024). In practice, we parameterize the posterior via the precision matrix $\mathbf{A}^k := (\boldsymbol{\Sigma}^k)^{-1}$ and store its Cholesky factor to ensure positive definiteness.

The trainable parameters $\{\boldsymbol{\theta}_h, \boldsymbol{\theta}_\phi\}$ of $h(\cdot)$ and $\boldsymbol{\phi}(\cdot)$ and all K regression heads are trained jointly by minimizing the weighted negative ELBO

$$\mathcal{L}(\{\boldsymbol{\Theta}_k\}_{k=1}^K, \boldsymbol{\theta}_h, \boldsymbol{\theta}_\phi) = \sum_{k=1}^K \frac{|\mathcal{S}_k|}{N} \cdot \big(-\mathcal{L}^{\text{ELBO}}(\boldsymbol{\Theta}_k)\big)\,. \tag{5}$$

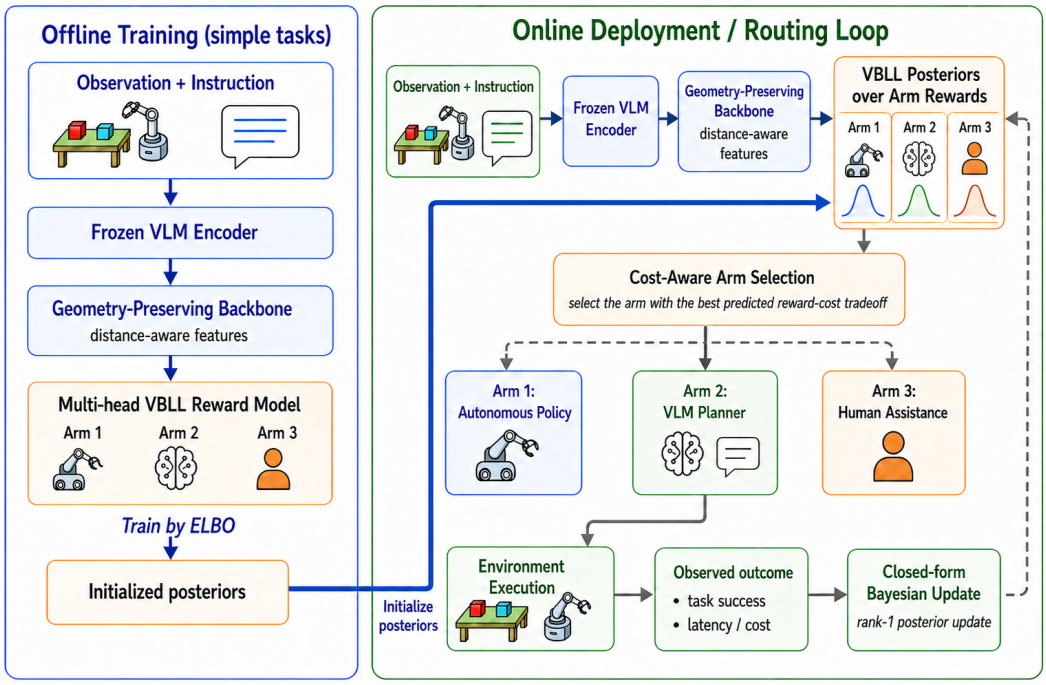


Figure 2: Overview of BARM. Left: offline training on simple tasks initializes the multi-head VBLL reward posteriors from frozen VLM features and a distance-aware backbone. Right: during online deployment, the router extracts context features, forms calibrated posterior beliefs over arm rewards, performs cost-aware arm selection, observes execution outcomes, and updates the selected arm by a closed-form Bayesian posterior update.

The training process is carried out offline using data collected from simple (in-distribution) tasks only. Composite tasks, whose optimal arm preferences differ, are encountered only during online evaluation, giving rise to the cold-start problem described in Section 2.

3.3 TS FOR ONLINE ARM SELECTION

At each episode t, the agent observes a task context (s_t, ℓ_t) and must commit to one arm/execution mode (note that we use index t/n for online/offline sample index). Following the Bayesian agentic control perspective of (Papamarkou et al., 2026), we treat the VBLL posterior as a belief state over the reward parameters and let the agent act greedily with respect to a sample drawn from this belief, which is the standard recipe of TS (Thompson, 1933; Daniel et al., 2018). Given the SNGP feature $\boldsymbol{\phi}_t$ from Eq. equation 9 and the current posterior parameters $\{\boldsymbol{\mu}_t^k, \boldsymbol{\Sigma}_t^k, \sigma_{\eta,k}^2\}$, this cost-aware TS works as

$$a_t = \arg\max_{k\in\{1,\dots,K\}} \tilde{r}_t^k - c_k, \text{ where } \tilde{r}_t^k = \hat{\boldsymbol{\beta}}_t^{k\top}\boldsymbol{\phi}_t, \;\; \hat{\boldsymbol{\beta}}_t^k \sim \mathcal{N}\big(\boldsymbol{\mu}_t^k, \;\; \alpha^2\boldsymbol{\Sigma}_t^k\big)\,, \tag{6}$$

where $\alpha \in (0, 1]$ is an exploration scale that modulates the posterior variance, and $c_k \geq 0$ is the known per-use cost of arm k. Since each weight vector is drawn from the current posterior, the probability that arm k is played coincides with the posterior probability that arm k is cost-optimal at $\boldsymbol{\phi}_t$, which is the probability-matching property of TS. As reward observations accumulate and the posterior concentrates, the same rule produces increasingly deterministic selections, so exploration gives way to exploitation without any change of mechanism.

3.4 CLOSED-FORM ONLINE BAYESIAN POSTERIOR UPDATE

After executing arm a_t and observing reward r_t, we refresh the posterior of the selected arm via a closed-form conjugate update in sufficient-statistics form, while parameters of unselected arms remain unchanged. Two design choices distinguish this update from the LinUCB rule of (Li et al.,

2010): the precision is initialized from the offline VBLL posterior $\hat{\mathbf{\Sigma}}^k$ rather than a ridge prior $\lambda\mathbf{I}$, so calibrated ELBO uncertainty carries into the online phase; and the posterior is paired with TS rather than UCB, enabling cost-aware probability matching. Each update is a rank-1 operation of cost $\mathcal{O}(D_L^2)$. Update equations, initialization formulas, and the full BARM algorithm are deferred to App. C.

Please refer to Algorithm 1 in APP. D for the details of the proposed method.

Remark 1 (BARM as an orchestration layer for VLAs). Although instantiation in the present work assigns the first arm to a diffusion policy, the routing framework is agnostic to the identity of its execution modes. If the first arm were replaced by a vision-language-action (VLA) model, BARM would function as a principled orchestration layer that decides when the VLA should act autonomously, when it should be augmented by VLM-based decomposition, and when a human operator should intervene. Thus, it could naturally serve as a complementary module to VLAs rather than a competitor. We leave this for our research agenda.

4 Regret Analysis

We now state a Bayesian regret bound for BARM. The full list of assumptions, the proof, and a parallel analysis of an SNGP+LinUCB variant are deferred to Appendix G, where Appendix G.1 collects the technical conditions used below.

Theorem 1 (Bayesian regret of BARM)**.** *Under the well-specified linear reward model and the regularity conditions stated in Appendix G.1,*

$$\mathrm{BayesRegret}(T) \;=\; \widetilde{O}\big(K D_L \sqrt{T}\big), \tag{7}$$

where K is the number of arms, D_L is the RFF feature dimension, T is the online horizon, and $\widetilde{O}$ hides factors logarithmic in T together with constants depending on the noise scale, the parameter norm bound, and the maximum arm cost.

The bound is sublinear in T, so the per-episode Bayes regret vanishes as the deployment horizon grows. As a sanity check, the same SNGP feature space also supports a frequentist confidence-bound analysis for an SNGP+LinUCB variant, which gives $\widetilde{O}(D_L\sqrt{KT})$ (Theorem 3). The two rates agree in their dependence on T and differ by a factor of $\sqrt{K}$, consistent with the well-known gap between TS and LinUCB analyses for linear contextual bandits (Agrawal & Goyal, 2013; Abbasi-Yadkori et al., 2011). This gap reflects a limitation of distribution-independent Bayes risk analysis rather than a real performance gap, and matches our experiments, where TS in fact outperforms UCB. The correspondence between the three components of BARM and the conditions used in the proof is laid out in Appendix G.5.

5 Experiments

We evaluate two central questions: whether assistance routing benefits from a Bayesian cost-aware decision formulation, and whether robustness under distribution shift requires the joint alignment of representation geometry, uncertainty calibration, and exploration, rather than any single ingredient in isolation.

5.1 Setup

BARM is evaluated on two language-conditioned robotic manipulation benchmarks: (1) CALVIN (Mees et al., 2022) with 28 tasks (13 simple, 15 composite), and (2) LIBERO with 32 tasks (20 simple, 12 composite). The diffusion policy is MoDE (Reuss et al., 2024); the VLM is LLaVA-OneVision-7B (Li et al., 2024). We consider three execution modes with heterogeneous per-use costs: arm 1 executes MoDE directly at the lowest cost (c_1=0); arm 2 uses VLM-based task decomposition with cost c_2=0.15 on CALVIN and c_2=0.10 on LIBERO; arm 3 requests human demonstration (c_3=0.30). Net reward per episode is the raw task reward minus the cost of the selected arm. Offline training uses simple tasks only (CALVIN: 390 samples; LIBERO: 600 samples), and composite tasks are unseen during training. Online evaluation runs 500 episodes

per seed (50% simple, 50% composite), averaged over 5 seeds. We compare BARM against 22 methods spanning four categories: detection, RL and learned routing, contextual bandit, and other neural baselines. All neural methods share the same frozen SNGP features and have access to online updates, isolating the routing mechanism as the sole variable. The full baseline list, instantiation details, and hyperparameters are deferred to Appendix E.1.

5.2 Main results

Table 1 reports the main results. BARM achieves the highest net reward on both benchmarks, outperforming all 22 baselines. The four categories of baselines fall short for distinct, complementary reasons.

Table 1: Net reward on CALVIN and LIBERO (mean ± standard deviation over five seeds). Best non-Oracle bolded.

Method	CALVIN	LIBERO
Oracle	.854±.012	.905±.005
Ours		
BARM	**.786**±.020	**.834**±.011
BARM(IDS (Russo & Van Roy, 2014))	.738±.024	.812±.012
BARM(w/o TS)	.724±.026	.804±.010
VBLL+TS (no SNGP)	.727±.006	.771±.004
Bandit methods		
Neural-LinUCB	.752±.008	.796±.005
EE-Net	.609±.021	.759±.016
NeuralTS	.590±.024	.709±.017
Detection methods		
SNGP+Conf	.720±.016	.697±.015
IntroPlan	.671±.013	.576±.017
FIPER	.669±.015	.648±.008
Sentinel	.661±.031	.706±.012
KnowNo	.608±.015	.691±.009
CogVLA	.603±.020	.671±.011
DeeR-VLA	.599±.019	.714±.012
RL & learned routing		
LLM Routing	.643±.018	.788±.024
MCDrop	.634±.037	.797±.009
FiS-VLA	.593±.020	.720±.015
DQN	.611±.034	.756±.018
VADER	.599±.015	.612±.022
AutoVLA	.597±.017	.688±.023
PPO	.596±.021	.600±.009
REINFORCE	.593±.023	.560±.017
Fixed policies		
Always Arm1	.593±.020	.489±.008
Always Arm3	.554±.015	.585±.004
Random	.545±.021	.558±.013
Always Arm2	.484±.021	.576±.017

Detection methods (SNGP+Conf, Sentinel, FIPER, IntroPlan, KnowNo, CogVLA, DeeR-VLA) are frozen at deployment by design: their thresholds are calibrated offline and do not adapt based on observed outcomes. SNGP+Conf performs best within this category by leveraging the same distance-aware features as BARM, but it does not model per-arm reward and therefore cannot reason about which mode yields the highest net reward for a given context. CogVLA and DeeR-VLA use different confidence signals (softmax entropy versus logistic regression) but face the same limitation: fixed thresholds cannot adapt to the per-task reward structure encountered online.

RL and learned routing methods (PPO, DQN, REINFORCE, AutoVLA, VADER, FiS-VLA) optimize reward through online updates but lack an explicit per-arm reward model. AutoVLA, despite combining SFT with GRPO, cannot learn the compositional-task routing pattern in 500 episodes because policy gradient signals are too sparse. VADER's online-updated affordance model degrades relative to its frozen variant, and FiS-VLA's gate collapses to arm 1 on CALVIN, both confirming that SGD-based online updates are destabilizing in the low-sample regime.

Bandit methods (Neural-LinUCB, NeuralTS, EE-Net) explicitly model per-arm reward with online updates and share the same SNGP features as BARM. Their consistent ordering, with Neural-LinUCB outperforming NeuralTS and EE-Net, indicates posterior quality as the differentiating factor: the exact ridge-regression update produces better-calibrated uncertainty than the gradient-based diagonal posteriors of NeuralTS or the dual heads of EE-Net. BARM further improves over Neural-LinUCB by replacing its flat ridge prior with a calibrated VBLL posterior trained via ELBO, with the improvement concentrated on OOD tasks (Section 5.3).

A counterpart of Table 1 in which all baselines use a plain MLP projection instead of SNGP features is reported in Table 4 (Appendix E.2). SNGP features are broadly beneficial across baselines, and the gap between BARM and the strongest competitor in Table 1 is therefore not attributable to the backbone alone.

5.3 Arm selection analysis

Figure 3 compares each method's arm selection accuracy against the Oracle. On simple tasks, nearly all methods achieve near-perfect accuracy, with FIPER as the notable exception. The key distinction arises in composite tasks, where the agent must discover through interaction that costlier and more capable arms are required. BARM attains the highest composite-task accuracy on both CALVIN (55%) and LIBERO (67%), substantially outperforming all baselines. Neural-LinUCB explores but less selectively, since its flat prior assigns similar uncertainty to ID and OOD inputs at the start of online evaluation. AutoVLA is locked onto arm 1 on CALVIN by SFT bias that GRPO cannot

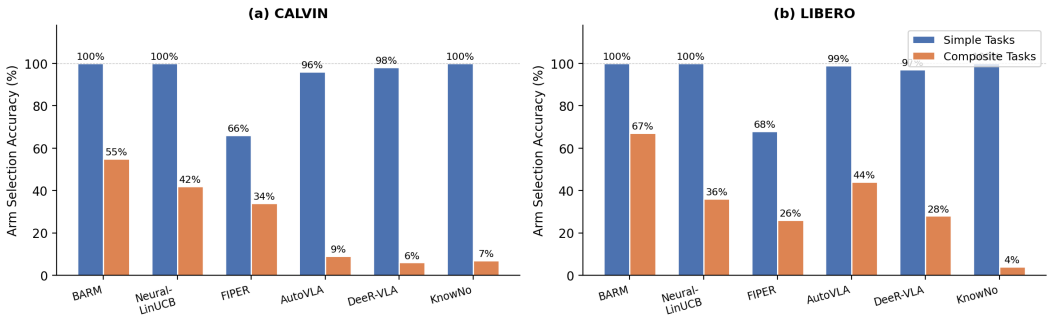


Figure 3: Arm selection accuracy versus Oracle on CALVIN (a) and LIBERO (b), split by simple and composite tasks. BARM achieves the highest composite-task accuracy on both benchmarks while maintaining near-perfect simple-task accuracy.

Table 2: Ablations on CALVIN. **(a)** Acquisition function (BARM backbone). **(b)** Feature backbone (all with TS). **(c)** Online update (BARM). **(d)** Execution latency.

(a) Acq. Function

Strategy	Reward
TS	**.786**
UCB	.738
IDS	.738
Greedy	.724

(b) Backbone

Backbone	Reward
BARM	**.786**
Neural-LinUCB	.752
MLP	.693
MCDrop	.634

(c) Online Update

	CAL	LIB
Bayes	**.786**	**.834**
SGD	.612	.796

(d) Latency

Method	s/ep
FIPER	3.83
BARM	**4.22**
Neural-LinUCB	4.56
MCDrop	5.14
IntroPlan	6.0
KnowNo	6.7

correct within the online horizon, while DeeR-VLA's fixed confidence thresholds and KnowNo's near-deterministic default to arm 1 yield similarly poor composite-task accuracy. These results confirm that calibrated posterior uncertainty paired with TS is uniquely effective at discovering cost-aware routing online.

5.4 Ablation Studies

Table 2 verifies the joint necessity of the three BARM components. Panel (a) shows that TS outperforms UCB, IDS, and Greedy with the BARM backbone fixed: Greedy's degradation is especially informative, since the calibrated posterior correctly elevates epistemic variance on composite tasks, but Greedy ignores this signal and defaults to arm 1 on OOD inputs. Panel (b) shows that, with TS fixed, the SNGP backbone outperforms Neural-LinUCB, plain MLP, and MCDropout features. Without spectral normalization, the MLP feature matrix's singular values collapse after the first 15 dimensions, compressing the feature space into a low-rank subspace where OOD and ID inputs become indistinguishable (Appendix E.4, Figure 7). Panel (c) shows that the closed-form Bayesian conjugate update outperforms online SGD by $+17.4$pp on CALVIN and $+3.8$pp on LIBERO: SGD overreacts to early composite-task failures and permanently distorts the posterior mean, while the rank-1 conjugate update preserves the structure of the offline-trained prior. Panel (d) reports per-episode latency: BARM is comparable to FIPER and faster than VLM-based detection baselines, since the rank-1 conjugate update adds negligible overhead relative to a single environment step. A counterpart of these ablations on raw VLM features is reported in Table 5 (App. E.3), and a per-difficulty decomposition of arm selection across ID, Borderline, and OOD bins is provided in Figure 6 (App. E.4).

5.5 Predictive Calibration

Table 3 evaluates predictive calibration independently of routing performance. BARM achieves the best NLL across all four conditions on CALVIN. The gap is largest on Offline OOD, where MLP+VBLL produces severely overconfident predictions on unseen

Table 3: Negative log-likelihood (NLL) on CALVIN (5 seeds).

Method	Off. ID	Off. OOD	On. ID	On. OOD
BARM	$\mathbf{-1.10}\pm.28$	$\mathbf{0.98}\pm.17$	$\mathbf{-1.32}\pm.15$	$\mathbf{-0.01}\pm.05$
Neural-LinUCB	$-1.02\pm.32$	$1.00\pm.13$	$-1.31\pm.10$	$0.09\pm.03$
MLP+VBLL	$-0.56\pm.25$	$2.93\pm.35$	$-1.30\pm.00$	$0.39\pm.10$
LLM Router	$1.44\pm.09$	$1.05\pm.08$	$2.45\pm.33$	$0.29\pm.05$
MCDrop	9.04 ± 1.35	$1.28\pm.14$	$5.08\pm.50$	$0.41\pm.05$

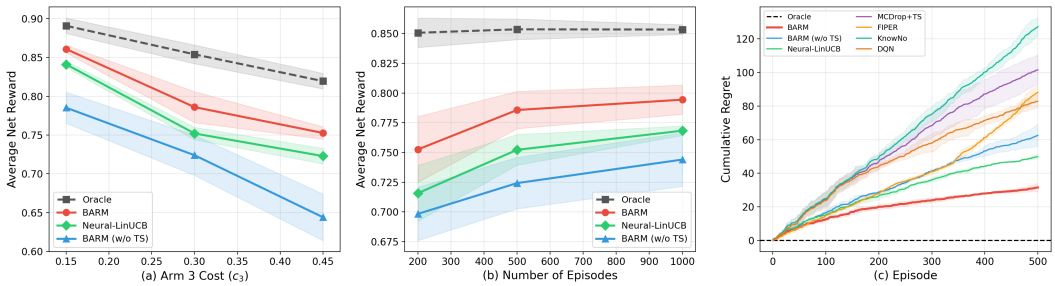


Figure 4: Sensitivity analysis on CALVIN. **(a)** Cost sensitivity. **(b)** Episode sensitivity. **(c)** Cumulative regret versus Oracle over 500 episodes.

composite tasks, confirming the feature-collapse mechanism described in Section 5.4. The ordering is preserved on the online splits as the posteriors absorb composite-task observations through the conjugate update of Eq. equation 10, indicating that BARM's routing advantage stems from better-calibrated predictive distributions: the SNGP backbone elevates epistemic variance on OOD inputs, which TS then translates into appropriate exploration of costlier arms.

5.6 Why SNGP features are necessary: a mechanism-level view

Figure 7 exposes the mechanism behind the SNGP-vs-MLP gap reported in Table 2(b). Panel (c) shows that without spectral normalization, the singular values of the MLP feature matrix decay to $\sim 10^{-11}$ after about 15 dimensions, collapsing the representation into a low-rank subspace in which OOD and ID inputs are no longer linearly separable. As a result, the predictive spread in panel (b) fails to distinguish composite from simple tasks, and any last-layer posterior on these features inherits a uniformly compressed uncertainty estimate. The SNGP backbone retains full rank, and its predictive variance in panel (a) tracks the prediction error and is elevated on composite tasks, supplying the distance-aware signal that VBLL and TS rely on.

5.7 Sensitivity analysis

Figure 4 evaluates robustness along three axes. BARM maintains its advantage over all baselines as the arm 3 cost varies from 0.15 to 0.45 (panel a), with the largest margin at moderate costs where the cost-benefit tradeoff is most nuanced. The advantage grows with longer deployment horizons (panel b), consistent with the Bayesian posterior accumulating evidence and enabling increasingly precise arm selection. Panel (c) shows that BARM accumulates the least regret against the Oracle, in line with the sublinear $\widetilde{O}(KD_L\sqrt{T})$ rate of Theorem 1, while RL and frozen detection methods diverge rapidly.

5.8 Additional Robustness Experiments

We further evaluate BARM under action-irrelevant support expansion (Table 7), a reverse distribution shift (Tables 8 and 9), reward-model misspecification (Tables 10 and 11), stochastic assistance costs and latency (Table 12), stochastic human availability (Table 13), and additional execution modes (Table 14).

6 Conclusion

We proposed BARM, a unified Bayesian framework whose single belief state simultaneously satisfies both desiderata: a distance-aware representation, multi-head VBLL posteriors trained offline by ELBO are updated online via closed-form rank-1 conjugate updates, and TS on the resulting posterior performs cost-aware arm selection. We complement the algorithm with a Bayesian regret analysis that yields a sublinear bound under a well-specified linear reward model. On CALVIN and LIBERO, BARM achieves the highest cost-adjusted reward against 22 baselines, and ablations confirm that removing either desideratum substantially degrades performance.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and manuscript organization.

ETHICS STATEMENT

This work does not involve human-subject experiments, sensitive personal data, or real-world deployment.

REPRODUCIBILITY STATEMENT

We provide the full algorithm in Appendix D, experimental details in Appendix E, additional robustness experiments in Appendix F, and complete assumptions and proofs in Appendix G.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Proc. of Adv. Neural Inf. Process. Syst.*, 24, 2011.

Christopher Agia, Rohan Sinha, Jingyun Yang, Zi-ang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. *arXiv preprint arXiv:2410.04640*, 2024.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proc. Intl. Conf. Mach. Learn.*, pp. 127–135. PMLR, 2013.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Michael Ahn, Montserrat Gonzalez Arenas, Matthew Bennice, Noah Brown, Christine Chan, Byron David, Anthony Francis, Gavin Gonzalez, Rainer Hessmer, Tomas Jackson, et al. Vader: Visual affordance detection and error recovery for multi robot human collaboration. *arXiv preprint arXiv:2405.16021*, 2024.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. Ee-net: Exploitation-exploration neural networks in contextual bandits. *arXiv preprint arXiv:2110.03177*, 2021.

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *Proc. Intl. Conf. Mach. Learn.*, 2015.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *Proc. Intl. Conf. Learn. Repr.*, 2019.

Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Proc. of Adv. Neural Inf. Process. Syst.*, 2011.

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *arXiv preprint arXiv:2506.01953*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Intl. J. Robot. Res.*, 44(10-11):1684–1704, 2025.

J Russo Daniel, Van Roy Benjamin, Kazerouni Abbas, Osband Ian, and Wen Zheng. A tutorial on thompson sampling. *Found. Trends Mach. Learn.*, 11(1):1–99, 2018.

Morris H DeGroot. *Optimal statistical decisions*. John Wiley & Sons, 2005.

Alec Farid, Sushant Veer, and Anirudha Majumdar. Task-driven out-of-distribution detection with statistical guarantees for robot learning. In *Proc. Conf. Robot Learn.*, pp. 970–980. PMLR, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Machine Learning Res.*, 23(120):1–39, 2022.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Intl. Conf. Mach. Learn.*, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proc. Intl. Conf. Mach. Learn.*, 2017.

James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers. In *Proc. Intl. Conf. Learn. Repr.*, 2024.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *Proc. IEEE Intl. Conf. Robot. Autom.*, 2011.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *Proc. IEEE Intl. Conf. Robot. Autom.*, pp. 8077–8083. IEEE, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. of Adv. Neural Inf. Process. Syst.*, 2017.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. Intl. Conf. World Wide Web*, pp. 661–670, 2010.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *arXiv preprint arXiv:2508.21046*, 2025.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *Proc. IEEE Intl. Conf. Robot. Autom.*, pp. 9493–9500. IEEE, 2023.

Kaiqu Liang, Zixu Zhang, and Jaime F Fisac. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. *Proc. of Adv. Neural Inf. Process. Syst.*, 37:71998–72031, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Proc. of Adv. Neural Inf. Process. Syst.*, 36:44776–44791, 2023.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Proc. of Adv. Neural Inf. Process. Syst.*, 2020.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robot. Autom. Lett.*, 7(3):7327–7334, 2022.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. Intl. Conf. Learn. Repr.*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Proc. of Adv. Neural Inf. Process. Syst.*, 2019.

Theodore Papamarkou, Pierre Alquier, Matthias Bauer, Wray Buntine, Andrew Davison, Gintare Karolina Dziugaite, Maurizio Filippone, Andrew YK Foong, Vincent Fortuin, Dimitris Fouskakis, et al. Position: Agentic ai systems should be making bayes-consistent decisions. 2026.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proc. of Adv. Neural Inf. Process. Syst.*, 2007.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. *arXiv preprint arXiv:2412.12953*, 2024.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018.

Herbert Robbins. Some aspects of the sequential design of experiments. 1952.

Ralf Römer, Adrian Kobras, Luca Worbis, and Angela P Schoellig. Failure prediction at runtime for generative robot policies. *arXiv preprint arXiv:2510.09459*, 2025.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Math. Oper. Res.*, 39(4):1221–1243, 2014.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *J. Machine Learning Res.*, 17(68):1–30, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

Bates Stephen et al. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv: 2107.07511*, 2021.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Abraham Wald. *Statistical decision functions*. Springer, 1950.

Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. Blob: Bayesian low-rank adaptation by backpropagation for large language models. In *Proc. of Adv. Neural Inf. Process. Syst.*, 2024.

Joe Watson, Jihao Andreas Lin, Pascal Klink, Joni Pajarinen, and Jan Peters. Latent derivative bayesian last layer networks. In *Proc. Intl. Conf. Artif. Intell. Stat.*, 2021.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. Neural contextual bandits with deep representation and shallow exploration. *arXiv preprint arXiv:2012.01780*, 2020.

Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. *Proc. of Adv. Neural Inf. Process. Syst.*, 37:56619–56643, 2024.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. *arXiv preprint arXiv:2010.00827*, 2020.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025.

Supplementary File

A Related Work

Uncertainty-aware planning and failure detection. Deciding when an autonomous system should seek help has been studied across planning and policy execution. Early work on safe robot learning introduced human-gated interventions triggered by hand-crafted heuristics (Kelly et al., 2019). More recent approaches leverage learned uncertainty signals. On the planning side, SayCan (Ahn et al., 2022) grounds VLM plans in learned affordance functions, Inner Monologue (Huang et al., 2022) uses environment feedback to trigger LLM-based replanning, and KnowNo (Ren et al., 2023) applies conformal prediction (Vovk et al., 2005; Stephen et al., 2021) over VLM logits to produce statistically calibrated ask-or-execute decisions. IntroPlan (Liang et al., 2024) further simplifies this by eliciting uncertainty through introspective prompting without additional calibration data. On the policy side, (Farid et al., 2022) derives PAC-Bayes bounds for OOD detection in policy deployment, Sentinel (Agia et al., 2024) monitors diffusion rollouts for internal consistency violations, and FIPER (Römer et al., 2025) applies Random Network Distillation (Burda et al., 2019) for computationally efficient novelty scoring. VADER (Ahn et al., 2024) combines affordance-based failure detection with a multi-level help-seeking hierarchy (retry, ask robot, ask human) that motivates our multi-arm formulation. These methods provide effective mechanisms for recognizing when autonomous execution is unreliable. Our work builds on this foundation and complements it by additionally modeling the expected downstream reward of each execution mode, enabling routing decisions that account for both input novelty and the cost-benefit tradeoff of escalation.

Learned routing and RL-based approaches. The idea of dynamically allocating compute to match task difficulty originates in mixture-of-experts architectures (Jacobs et al., 1991), where gating networks learn to route inputs to specialized sub-networks. Modern instantiations such as (Shazeer et al., 2017) and Switch Transformer (Fedus et al., 2022) scale this to large models via sparse token-level routing. In embodied AI, MoDE (Reuss et al., 2024) extends mixture-of-experts to diffusion policies with noise-conditioned gating, DeeR-VLA (Yue et al., 2024) introduces confidence-based early exit to reduce inference cost, CogVLA (Li et al., 2025) prunes tokens and layers via instruction-driven gates, and FiS-VLA (Chen et al., 2025) implements a dual-system split between fast and slow pathways. These methods achieve efficient inference by adapting computation to task complexity. Going beyond compute allocation, RL-based methods optimize routing for task reward directly: standard algorithms such as PPO (Schulman et al., 2017), DQN (Mnih et al., 2015), and REINFORCE (Williams, 1992) can learn routing policies from completion signals, and AutoVLA (Zhou et al., 2025) applies GRPO to learn when to invoke chain-of-thought reasoning in autonomous driving. While these approaches effectively incorporate reward optimization, the sparse reward signal and the absence of an explicit per-mode reward model can limit sample efficiency in the low-data regime typical of robotic deployment, where only a few hundred online episodes are available. Our approach addresses this by maintaining explicit per-arm reward posteriors with closed-form Bayesian updates, enabling sample-efficient adaptation without gradient-based policy optimization.

Contextual bandits and Bayesian decision theory. Selecting actions under uncertainty to maximize expected utility is a classical problem in statistical decision theory (Wald, 1950; Berger, 2013; DeGroot, 2005). Its sequential, context-dependent extension, the contextual bandit (Robbins, 1952; Auer et al., 2002; Li et al., 2010), provides a principled framework for balancing exploration and exploitation when the agent observes side information before acting. LinUCB (Li et al., 2010) maintains closed-form ridge-regression posteriors with upper confidence bound exploration, and TS (Thompson, 1933) achieves probability-matching exploration with sublinear Bayes regret (Daniel et al., 2018). To address the representational limitations of linear reward models, neural extensions have been proposed: Neural-LinUCB (Xu et al., 2020) periodically retrains learned features, NeuralTS (Zhang et al., 2020) employs diagonal gradient-based posteriors over neural network parameters, and EE-Net (Ban et al., 2021) separates exploitation and exploration into dual prediction heads. The comprehensive benchmark of (Riquelme et al., 2018) demonstrates that Bayesian inference on learned features achieves the best uncertainty–computation tradeoff, motivating our design. These methods have been primarily developed for settings with homogeneous arm costs and stationary feature distributions, such as news recommendation and online advertising. Our work extends the contextual bandit framework to embodied routing, where arm costs are heterogeneous and a cold-start

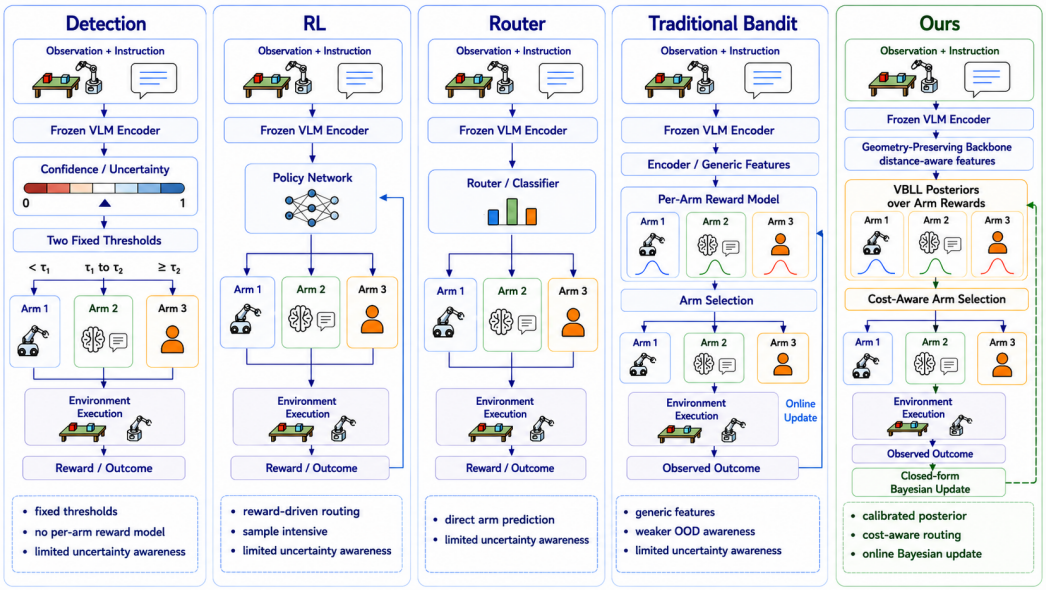


Figure 5: Conceptual comparison of representative routing paradigms for embodied assistance. Detection methods rely on fixed uncertainty thresholds; RL methods optimize routing through reward-driven policy learning; discriminative routers directly predict the arm; traditional bandit methods model per-arm rewards but typically use weaker features and uncertainty estimates. BARM combines distance-aware features, calibrated VBLL posteriors over arm rewards, cost-aware arm selection, and closed-form Bayesian online updates.

problem arises because the offline prior covers only atomic tasks while deployment involves unseen compositional instructions. (Papamarkou et al., 2026) recently argue that agentic AI orchestration should be Bayes-consistent and sketch a TS routing example in their Appendix C; we provide the first empirical instantiation and identify the feature backbone as the binding constraint.

Distance-aware uncertainty estimation. Reliable uncertainty quantification is essential for Bayesian decision-making, yet neural networks are often poorly calibrated (Guo et al., 2017), and calibration can degrade further under distribution shift (Ovadia et al., 2019). Early approaches to this problem include MC-Dropout (Gal & Ghahramani, 2016), which reinterprets dropout at test time as approximate Bayesian inference, and Deep Ensembles (Lakshminarayanan et al., 2017), which aggregate predictions from independently trained models. Variational Bayesian methods such as Bayes by Backprop (Blundell et al., 2015) provide a more principled framework by learning weight posteriors during training; BLoB (Wang et al., 2024) extends this to parameter-efficient fine-tuning of LLMs, demonstrating that jointly optimizing the mean and covariance of variational distributions yields superior calibration compared to post-training Bayesianization. For last-layer inference specifically, (Watson et al., 2021) and VBLL (Harrison et al., 2024) restrict Bayesian inference to the output layer for tractability while preserving calibrated predictive distributions. A key insight formalized by (Liu et al., 2020) is that the calibration of any last-layer method depends on two conditions: a distance-preserving hidden mapping, enforced via spectral normalization (Miyato et al., 2018), and a distance-aware output layer with a shift-invariant kernel (Rahimi & Recht, 2007). Without the former, the hidden representation can collapse, rendering the posterior uninformative regardless of the quality of inference. We inherit this two-condition framework and extend it to the bandit setting, where the consequences of miscalibration are qualitatively different from those in passive prediction: rather than degrading a confidence estimate, it can misdirect the exploration policy that determines how the agent allocates costly resources across execution modes.

B DISTANCE-AWARE FEATURE BACKBONE VIA SNGP

The contextual bandit requires estimating per-arm reward uncertainty from features, but standard MLPs can map semantically dissimilar inputs to nearby points in feature space (*feature collapse*), producing overconfident posteriors on OOD inputs regardless of the downstream inference method (Liu et al., 2020). In our setting, this means compositional tasks unseen during training could appear indistinguishable from simple tasks in feature space, preventing the posterior from recognizing that exploration is needed. We adopt SNGP (Liu et al., 2020) to obtain *input distance-aware* features through two mechanisms.

Distance-preserving hidden mapping. We parameterize the feature backbone as a residual MLP $h(\cdot) = h_{L-1} \circ \cdots \circ h_1(\cdot)$ with layers $h_l(\mathbf{x}) = \mathbf{x} + g_l(\mathbf{x})$. Applying spectral normalization (Miyato et al., 2018) to constrain $\|g_l\|_{\text{Lip}} \leq c < 1$ ensures $h(\cdot)$ is bi-Lipschitz (Liu et al., 2020):

$$(1-c)^{L-1}\|\mathbf{x}-\mathbf{x}'\| \;\leq\; \|h(\mathbf{x}) - h(\mathbf{x}')\| \;\leq\; (1+c)^{L-1}\|\mathbf{x}-\mathbf{x}'\|, \tag{8}$$

Without spectral normalization, the feature matrix's singular values collapse to $\sim 10^{-11}$ after ~ 15 dimensions (Figure 7(c)), rendering the representation unable to distinguish OOD from in-distribution inputs.

Distance-aware output layer. We replace the dense output layer with a Random Fourier Feature (RFF) layer (Rahimi & Recht, 2007) that approximates an RBF-kernel Gaussian process. The RFF mapping $\boldsymbol{\phi}(\cdot) : \mathbb{R}^{D_{L-1}} \to \mathbb{R}^{D_L}$ uses fixed random projections, and the resulting kernel's posterior variance grows monotonically with distance from the training data. The full SNGP feature is

$$\boldsymbol{\phi}_t := \boldsymbol{\phi}(h(\mathbf{x}_t)) \in \mathbb{R}^{D_L}, \tag{9}$$

shared across all K reward heads.

C CLOSED-FORM ONLINE BAYESIAN POSTERIOR UPDATE

This appendix provides the full details of the online posterior update introduced in Section 3.4 of the main text.

After executing arm a_t at episode t and observing reward r_t, we update the posterior for the selected arm using the conjugate Bayesian linear regression update in sufficient-statistics form. Let $\mathbf{A}_t^k := (\boldsymbol{\Sigma}_t^k)^{-1}$ denote the precision matrix and $\mathbf{h}_t^k := \mathbf{A}_t^k \boldsymbol{\mu}_t^k$ the information vector for arm k at episode t. The update for the selected arm a_t is

$$\mathbf{A}_{t+1}^{a_t} = \mathbf{A}_t^{a_t} + \sigma_{\eta,a_t}^{-2}\, \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\top, \quad \mathbf{h}_{t+1}^{a_t} = \mathbf{h}_t^{a_t} + \sigma_{\eta,a_t}^{-2}\, r_t\, \boldsymbol{\phi}_t, \quad \boldsymbol{\mu}_{t+1}^{a_t} = (\mathbf{A}_{t+1}^{a_t})^{-1}\mathbf{h}_{t+1}^{a_t}, \tag{10}$$

while the parameters for unselected arms remain unchanged: $\mathbf{A}_{t+1}^k = \mathbf{A}_t^k$ and $\mathbf{h}_{t+1}^k = \mathbf{h}_t^k$ for $k \neq a_t$. Each observation contributes a rank-1 term to the precision, weighted by the inverse noise variance σ_{η,a_t}^{-2} learned during offline ELBO training (Eq. equation 4). The posterior at $t = 0$ is initialized from the offline-trained estimates, $\mathbf{A}_0^k = (\hat{\boldsymbol{\Sigma}}^k)^{-1}$ and $\boldsymbol{\mu}_0^k = \hat{\boldsymbol{\mu}}^k$ for all k.

This update shares the sufficient-statistics form of LinUCB (Li et al., 2010) but differs in two ways relevant to our setting. The precision matrix is initialized from the offline-trained VBLL posterior $\hat{\boldsymbol{\Sigma}}^k$ rather than from a ridge prior $\lambda\mathbf{I}$, so the calibrated uncertainty learned during ELBO training is carried into the online phase instead of being discarded. The posterior is also paired with Thompson Sampling rather than a UCB rule, allowing arm selection to adapt to the heterogeneous cost structure through probability matching. Each update is a rank-1 operation of cost $\mathcal{O}(D_L^2)$ and is negligible relative to a single environment step.

D BARM ALGORITHM

Algorithm 1 summarizes the full BARM procedure described in Section 3. The offline phase (line 1) trains the SN-MLP backbone, the RFF projection, and the K VBLL heads jointly by minimizing the weighted negative ELBO of Eq. equation 5 on simple-task data $\mathcal{D}_{\text{simple}}$. The online posterior is

initialized from the offline VBLL posterior (line 2), so the calibrated uncertainty learned during ELBO training is carried into deployment rather than being replaced by a flat ridge prior. At each episode (lines 3–12), the agent extracts an SNGP feature, draws one weight sample per arm from the current scaled posterior, selects the arm with the highest sampled net reward, and refreshes the precision and information vector of the selected arm via the rank-1 conjugate update of Eq. equation 10. Each online step costs $\mathcal{O}(KD_L^2)$ for arm scoring plus a single rank-1 update for the played arm, both negligible relative to environment execution.

Algorithm 1 BARM

Require: Offline data $\mathcal{D}_{\text{simple}} = \{(\mathbf{x}_n, a_n, r_n)\}_{n=1}^N$, arm costs $\{c_k\}_{k=1}^K$, exploration scale α

1: **Offline training:** Train SN-MLP $h(\cdot)$, RFF $\boldsymbol{\phi}(\cdot)$, and K VBLL heads via Eq. equation 5; obtain $\{\hat{\boldsymbol{\mu}}^k, \hat{\boldsymbol{\Sigma}}^k, \hat{\sigma}^2_{\eta,k}\}_{k=1}^K$
2: Initialize $\mathbf{A}_0^k \leftarrow (\hat{\boldsymbol{\Sigma}}^k)^{-1},\ \mathbf{h}_0^k \leftarrow \mathbf{A}_0^k \hat{\boldsymbol{\mu}}^k,\ \boldsymbol{\mu}_0^k \leftarrow \hat{\boldsymbol{\mu}}^k$ for all k
3: **for** $t = 1, \ldots, T$ **do**
4: Observe (s_t, ℓ_t); extract $\mathbf{x}_t = \text{VLM}(s_t, \ell_t)$
5: Compute SNGP feature $\boldsymbol{\phi}_t$ via Eq. equation 9
6: **for** $k = 1, \ldots, K$ **do**
7: Sample $\hat{\boldsymbol{\beta}}_t^k \sim \mathcal{N}(\boldsymbol{\mu}_t^k, \alpha^2 \boldsymbol{\Sigma}_t^k)$; compute $\tilde{r}_t^k = \hat{\boldsymbol{\beta}}_t^{k\top} \boldsymbol{\phi}_t$
8: **end for**
9: Select a_t via Eq. equation 6
10: Execute arm a_t; observe reward r_t
11: Update $\mathbf{A}_t^{a_t}, \mathbf{h}_t^{a_t}, \boldsymbol{\mu}_t^{a_t}$ via Eq equation 10
12: **end for**
13: **return** Cumulative net reward $\sum_{t=1}^T (r_t - c_{a_t})$

E Additional Experimental Results

This appendix provides the experimental details and supporting analyses summarized in Section 5. We list the 22 baselines and their instantiations (Section E.1), provide the MLP-backbone counterpart of Table 1 (Section E.2), the raw-VLM-feature counterpart (Section E.3), and a per-task-difficulty decomposition together with the feature-quality visualization that accompanies the ablation in Section 5.4 (Section E.4).

E.1 Baselines

We compare BARM against 20 methods across four categories. (1) **Detection methods** that route based on confidence or novelty without modeling per-arm reward: KnowNo (Ren et al., 2023), IntroPlan, FIPER (Römer et al., 2025), DeeR-VLA (Yue et al., 2024), CogVLA (Li et al., 2025), Sentinel (Agia et al., 2024), and SNGP+Conf. DeeR-VLA performs confidence-based early exit within a VLA model; we instantiate this as hierarchical two-threshold routing where high VLM confidence routes to arm 1 (direct execution, analogous to a shallow exit), medium confidence to arm 2 (VLM decomposition, analogous to a mid-depth exit), and low confidence to arm 3 (human demonstration, analogous to a full-depth exit). CogVLA uses instruction-conditioned softmax gating; we instantiate this as a softmax classifier on SNGP features with entropy-based routing. Sentinel monitors internal consistency of policy rollouts; we instantiate this as an autoencoder on SNGP features, where high reconstruction error (anomaly) triggers escalation. (2) **RL and learned routing methods** that optimize reward through online updates: PPO, DQN, REINFORCE, AutoVLA (Zhou et al., 2025), VADER (Ahn et al., 2024), and FiS-VLA (Chen et al., 2025). AutoVLA applies SFT, followed by GRPO; we instantiate this as SFT on offline oracle-arm labels, followed by online GRPO fine-tuning. VADER uses affordance-based sequential escalation (retry $\rightarrow$ ask robot $\rightarrow$ ask human); we instantiate this with an online-updated affordance model and three-level escalation thresholds. FiS-VLA uses a fast/slow dual-system gate; we extend this to a 3-path gate (fast/medium/slow $=$ arm 1/2/3) with online SGD updates. (3) **Bandit methods** that model per-arm reward explicitly: Neural-LinUCB (Li et al., 2010), NeuralTS (Zhang et al., 2020), and EE-Net (Ban et al., 2021). (4) **Other neural methods**: LLM Routing (MLP with greedy selection and online SGD) and

MCDropout (Gal & Ghahramani, 2016). An Oracle upper bound runs all arms and takes the best net reward. We also report ablations: BARM(w/o SNGP), BARM(w/o TS), and BARM(IDS).

In our implementation, we use the LLaVA-OneVision-7B (Li et al., 2024) model and extract the hidden state of the last token from the final transformer layer, yielding $D_0 = 3584$. We adopt last-token pooling rather than mean pooling because the final token attends to the full context under causal attention, and therefore encodes the joint interaction between the scene and the instruction that drives task difficulty.

E.2 MLP-BACKBONE COUNTERPART OF TABLE 1

Table 4 evaluates all baselines when using a standard MLP projection (without spectral normalization) instead of SNGP features. BARM retains its SNGP backbone. Oracle and the fixed policies (Always Arm$k$, Random) do not depend on the projection and are reported with the same values as in Table 1. This isolates the effect of the SNGP backbone on baseline performance.

Table 4: BARM with SNGP backbone versus all baselines with MLP projection (5 seeds). Oracle and fixed policies are projection-independent and reproduced from Table 1 for reference.

Method	CALVIN	LIBERO
Oracle	0.854 ± 0.012	0.905 ± 0.005
Ours (SNGP backbone)		
BARM	$\mathbf{0.786 \pm 0.020}$	$\mathbf{0.834 \pm 0.011}$
Bandit methods (MLP projection)		
Neural-LinUCB	0.715 ± 0.018	0.727 ± 0.023
NeuralTS	0.642 ± 0.007	0.671 ± 0.014
EE-Net	0.596 ± 0.022	0.550 ± 0.013
Detection methods (MLP projection)		
FIPER	0.656 ± 0.021	0.642 ± 0.009
Sentinel	0.645 ± 0.019	0.689 ± 0.010
CogVLA	0.639 ± 0.042	0.668 ± 0.012
KnowNo	0.581 ± 0.041	0.642 ± 0.005
IntroPlan	0.593 ± 0.018	0.489 ± 0.006
DeeR-VLA	0.577 ± 0.019	0.697 ± 0.012
RL & learned routing (MLP projection)		
DQN	0.648 ± 0.017	0.663 ± 0.025
MCDrop	0.642 ± 0.013	0.597 ± 0.015
LLM Routing	0.557 ± 0.056	0.555 ± 0.009
FiS-VLA	0.593 ± 0.018	0.618 ± 0.090
AutoVLA	0.594 ± 0.017	0.568 ± 0.026
VADER	0.577 ± 0.016	0.592 ± 0.049
PPO	0.578 ± 0.017	0.560 ± 0.015
REINFORCE	0.562 ± 0.024	0.544 ± 0.011
Fixed policies		
Always Arm1	0.593 ± 0.020	0.489 ± 0.008
Always Arm3	0.554 ± 0.015	0.585 ± 0.004
Random	0.545 ± 0.021	0.558 ± 0.013
Always Arm2	0.484 ± 0.021	0.576 ± 0.017

The comparison between Tables 1 and 4 shows that SNGP features are broadly beneficial for the baselines themselves: the majority of methods achieve higher net reward with SNGP than with the MLP projection, with the effect more pronounced on LIBERO than on CALVIN, and the largest gains accruing to methods that rely heavily on the geometry of the feature space. The distance-aware features produced by SNGP are therefore not specific to BARM, but form a generally useful representation for cost-aware routing under the atomic-to-compositional distribution shift. Since the competing methods in Table 1 are already evaluated on the same SNGP features, the gap between BARM and the second-best method there cannot be explained by the feature backbone alone; it reflects the contribution of the calibrated VBLL posterior coupled with TS.

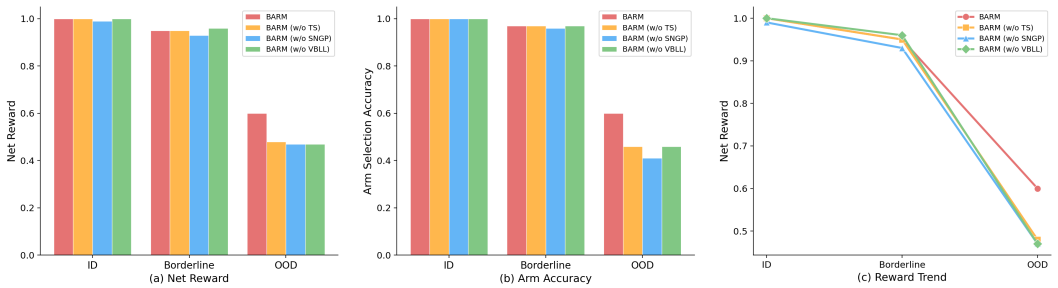


Figure 6: Arm selection behavior by task difficulty on CALVIN. Tasks are binned into ID (simple), Borderline, and OOD (composite). **(a)** Net reward per bin. **(b)** Arm selection accuracy versus Oracle. **(c)** Reward trend across bins. All methods perform comparably on ID tasks; the gap emerges on OOD tasks where BARM maintains the highest arm accuracy through calibrated exploration.

E.3 Raw-VLM-feature counterpart

Table 5 compares detection and learned routing baselines when using raw VLM features ($\mathbf{x}_t \in \mathbb{R}^{3584}$) instead of SNGP-projected features. The main-text results use SNGP features for all methods to isolate the routing mechanism; this table evaluates whether the baselines benefit from or depend on the SNGP-based backbone.

Table 5: Baselines with SNGP features (main text) versus raw VLM features. Our method (BARM) requires the SNGP-based backbone by design.

	CALVIN		LIBERO	
Method	SNGP feat.	Raw feat.	SNGP feat.	Raw feat.
BARM (ours)	**0.786**	0.701	**0.834**	0.725
VADER	0.599	0.572	0.612	0.548
Sentinel	0.661	0.725	0.706	0.416
FiS-VLA	0.593	0.693	0.720	0.725
CogVLA	0.603	0.580	0.671	0.580
DeeR-VLA	0.599	0.599	0.714	0.714

The results show that SNGP features do not uniformly help detection baselines: Sentinel and FiS-VLA perform better with raw features on CALVIN but worse on LIBERO, while VADER and CogVLA consistently degrade with raw features. This inconsistency across benchmarks highlights that detection methods are sensitive to the feature space in unpredictable ways. In contrast, BARM requires and fully exploits the SNGP backbone: distance-aware features are not merely helpful but architecturally necessary for the VBLL posterior to produce calibrated uncertainty that drives effective TS exploration.

E.4 Per-task-difficulty decomposition and feature-quality visualization

Figure 6 decomposes performance into ID, Borderline, and OOD bins on CALVIN. On ID (simple) tasks, all methods with reasonable posteriors achieve comparable rewards because the offline prior is already informative. The methods diverge on OOD (composite) tasks, where BARM maintains the highest arm selection accuracy. Methods without distance-aware features (VBLL-only+TS) and methods without posterior-driven exploration (BARM (w/o TS)) exhibit degraded arm accuracy on OOD tasks comparable to that of detection baselines. This confirms that the gains of BARM concentrate on unseen task types that require discovering new arm preferences through calibrated exploration.

Figure 7 substantiates the feature-collapse claim made in Section 5.4. Without spectral normalization, the MLP feature matrix's singular values collapse after the first 15 dimensions, compressing the

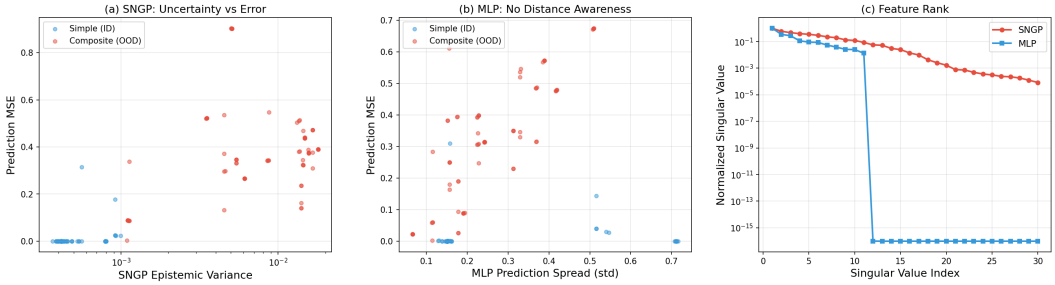


Figure 7: Feature quality comparison on CALVIN. **(a)** SNGP backbone: epistemic variance correlates with prediction error, with OOD (composite) tasks showing elevated variance. **(b)** Plain MLP backbone: prediction spread does not separate ID from OOD inputs. **(c)** Singular value spectra of the feature matrices: the MLP spectrum collapses to $\sim 10^{-11}$ after roughly 15 dimensions, while SNGP retains full rank.

Table 6: Action-irrelevant support expansion for one 7/6 seen–censored partition on CALVIN.

Method	Net reward	Δp_1 (pp)
BARM (TS)	0.9704 ± 0.0015	$+0.5 \pm 0.2$
Neural-LinUCB	0.9638 ± 0.0018	$+5.2 \pm 0.2$

feature space into a low-rank subspace where OOD and ID inputs become indistinguishable (panel c). The VBLL posterior receives features that carry no distance information and produces uncertainty estimates that fail to differentiate composite from simple tasks (panel b). The SNGP backbone, by contrast, maintains full feature rank, and its predictive variance correlates with prediction error and is elevated on OOD tasks (panel a). This confirms that a calibrated posterior is only as good as the features it receives, and that spectral normalization is the binding mechanism for distance-aware uncertainty in our setting. Removing TS while preserving SNGP and VBLL still degrades performance, since the SNGP+VBLL posterior correctly elevates epistemic variance on composite tasks, but Greedy ignores this signal and defaults to arm 1 on OOD inputs whose mean was trained only on simple tasks. Replacing MCDropout for VBLL while keeping SNGP and TS reproduces the failure mode of frozen detection baselines: dropout-based uncertainty does not correlate with distance from the training distribution, so TS draws from a posterior that assigns similar variance to ID and OOD inputs, resulting in undirected exploration.

F Additional Robustness Experiments

All experiments in this section are conducted on CALVIN, with online routing results averaged over five seeds.

F.1 Action-Irrelevant Support Expansion

We split the 13 atomic tasks into seven seen tasks used for offline reward-model training and six censored tasks hidden during training. At deployment, the router encounters all 13 tasks in a stable mixture, with Arm 1 optimal for every task. We report $\Delta p_1 = p_{1,\text{seen}} - p_{1,\text{censored}}$, where p_1 is the percentage of Arm 1 selections.

Table 6 shows that, for this partition, BARM changes its Arm 1 selection rate by only 0.5 percentage points between seen and censored tasks.

We repeat the experiment over five different 7/6 seen–censored partitions.

Across the five partitions in Table 7, BARM shows no systematic routing away from Arm 1 when the context distribution changes but the optimal action distribution remains fixed.

Table 7: Action-irrelevant support expansion over five different 7/6 seen–censored partitions on CALVIN.

Method	Net reward	Δp_1 (pp)
BARM (TS)	0.9714 ± 0.0010	-0.16 ± 1.73
Neural-LinUCB	0.9660 ± 0.0017	$+10.48 \pm 2.18$

Table 8: Net reward under the composite-to-atomic distribution shift on CALVIN.

Condition	Oracle	BARM (TS)	Neural-LinUCB
Composite $\rightarrow$ atomic	0.9757	0.9109	0.8200

F.2 Reverse Distribution Shift

The reward model is trained only on composite tasks, where the offline data indicate that Arm 1 performs poorly and favor the more capable modes. It is then deployed on atomic tasks, where Arm 1 is optimal.

We additionally report the Arm 1 selection percentage in consecutive 100-episode windows.

Table 8 shows that BARM achieves a higher net reward than Neural-LinUCB under the composite-to-atomic shift. As shown in Table 9, its Arm 1 selection rate rises from 57.0% in the first 100 episodes to approximately 80% thereafter, whereas Neural-LinUCB rises from 42.8% to 57.4%. BARM therefore adapts when the shift occurs in the opposite direction and the offline reward model initially favors the wrong arm.

F.3 Reward-Model Misspecification

For each arm, linear and MLP heads are fit on the same SNGP features to predict the per-task Monte Carlo expected reward, estimated from 30 cached trials per task.

Table 10 shows that the largest sup-norm residual is 0.045 for the VLM arm, while replacing the linear head with an MLP reduces the residual by at most 0.011. We also observe no systematic increase in approximation error from atomic to composite tasks.

We next construct a controlled sensitivity test,

$$r_\epsilon(x, a) = \langle \phi(x), \theta_a^* \rangle + \epsilon g_a(x),$$

where each residual component g_a is orthogonalized to the linear feature span and therefore cannot be removed by refitting the linear last layer.

As shown in Table 11, at $\epsilon = 0.05$, which approximately matches the largest measured residual of 0.045, BARM retains nearly all of its well-specified performance and remains ahead of Neural-LinUCB. Increasing the irreducible residual progressively reduces this advantage: BARM remains ahead at $\epsilon = 0.07$ and becomes comparable to Neural-LinUCB at $\epsilon = 0.08$, while both adaptive routers remain above the best fixed arm.

F.4 Stochastic Assistance Costs and Latency

The mean human cost remains 0.30. For cognitive-burden variation, the realized per-episode cost follows a Gaussian model around this mean. For response latency, it follows a heavy-tailed lognormal model with the same mean. BARM selects using the mean cost and is evaluated using the realized cost.

Table 12 shows that, across the tested range, net reward differs from the deterministic-cost result by at most 0.015, including under the heavy-tailed latency model. Thus, the observed routing performance does not rely on a deterministic realized human cost.

Table 9: Arm 1 selection percentage over deployment under the composite-to-atomic distribution shift.

Method	0–100	100–200	200–300	300–400	400–500
BARM (TS)	57.0	74.2	79.0	80.6	80.0
Neural-LinUCB	42.8	48.6	49.8	51.0	57.4

Table 10: Approximation error of the linear reward model on CALVIN. Brackets report Monte Carlo standard errors.

Arm	$\hat{\epsilon}_{\sup}$ (ID)	$\hat{\epsilon}_{\sup}$ (OOD)	$\hat{\epsilon}_{\mathrm{RMSE}}$	Linear-to-MLP reduction
Arm 1	0.016 [0.009]	0.015 [0.014]	≤ 0.009	≤ 0.001
Arm 2 (VLM)	0.037 [0.005]	0.045 [0.007]	≤ 0.019	≤ 0.011
Arm 3 (human)	0.024 [0.003]	0.015 [0.016]	≤ 0.010	≤ 0.004

F.5 Stochastic Human Availability

Arm 3 is independently unavailable with probability p at each episode and is removed from the feasible arm set before routing. The Oracle is subject to the same availability constraint.

As shown in Table 13, the achievable net reward decreases as human assistance becomes less frequently available. Relative to the Oracle under the same feasible arm set, BARM maintains a gap between 0.066 and 0.083 across the tested availability rates.

F.6 Additional Execution Modes

We add two execution modes only at deployment, with no offline observations, high fixed costs, and poor rewards. The interaction budget is unconstrained, so changes in their selection rates arise from online learning rather than forced pruning.

Table 14 shows that BARM does not hard-eliminate arms under posterior sampling, but rapidly suppresses both additional modes: each falls below 2% selection in the second 100-episode window and below 1% from the third window onward. Thus, the initially uncertain but consistently poor modes incur only limited early exploration before their low cost-adjusted value is learned.

G Bayesian Regret Analysis: Full Proofs

This appendix provides complete proofs of the Bayesian regret bounds. We restrict the entire analysis to the *well-specified regime*. We analyze two algorithms applied to the cost-adjusted contextual bandit of Section 2: (i) BARM (BARM, Algorithm 1), and (ii) SNGP+LinUCB. Both bounds achieve $\widetilde{O}(\sqrt{T})$ rates.

G.1 Assumptions

Assumption 1 (SNGP Bi-Lipschitz Mapping)**.** *The SNGP hidden mapping $h : \mathcal{X} \to \mathbb{R}^{D_{L-1}}$ satisfies*

$$\mu_{\mathrm{low}}\|\mathbf{x} - \mathbf{x}'\| \;\leq\; \|h(\mathbf{x}) - h(\mathbf{x}')\| \;\leq\; \mu_{\mathrm{up}}\|\mathbf{x} - \mathbf{x}'\| \qquad \forall\, \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \tag{11}$$

for constants $0 < \mu_{\mathrm{low}} \leq \mu_{\mathrm{up}} < \infty$. In particular, h is injective on $\mathcal{X}$.

Assumption 2 (Sub-Gaussian Noise)**.** *The reward noise $\xi_t := r_t - f^{a_t}_{\mathrm{true}}(\mathbf{x}_t)$ is conditionally σ-sub-Gaussian. Since $r_t \in [0, 1]$, Hoeffding's lemma gives $\sigma = 1/2$ under the convention $\mathbb{E}[\exp(\lambda\xi_t) \mid \mathcal{F}_{t-1}] \leq \exp(\sigma^2\lambda^2/2)$.*

Assumption 3 (Bounded Features)**.** *The RFF mapping satisfies $\boldsymbol{\phi}(z) = \sqrt{2/D_L}\cos(W_L z + b_L)$, so $\|\boldsymbol{\phi}_t\| \leq C_\phi := \sqrt{2}$ for all t and any D_L.*

Assumption 4 (Linear Realizability with Norm Bound)**.** *There exist (random) coefficients $\{\boldsymbol{\beta}^{k*}\}_{k=1}^{K}$ in $\mathbb{R}^{D_L}$ with $\|\boldsymbol{\beta}^{k*}\| \leq B$ almost surely, such that*

$$f^k_{\mathrm{true}}(\mathbf{x}) \;=\; \boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}(h(\mathbf{x})), \qquad \forall\, \mathbf{x} \in \mathcal{X},\; k = 1, \ldots, K. \tag{12}$$

Table 11: Sensitivity to controlled reward-model misspecification on CALVIN.

Injected ϵ	BARM (TS)	Neural-LinUCB	Best fixed arm
0.00	0.786	0.732	0.594
0.05	0.780	0.731	0.594
0.07	0.743	0.732	0.594
0.08	0.729	0.731	0.594

Table 12: BARM net reward under stochastic human-assistance costs on CALVIN.

Cost variation	Gaussian cognitive-burden cost	Lognormal response latency
$\sigma = 0.00$	0.776	0.776
$\sigma = 0.05$	0.774	0.774
$\sigma = 0.10$	0.784	0.781
$\sigma = 0.20$	0.763	0.761

We refer to this as the well-specified regime.

Remark (On the well-specified regime)**.** *We assume that, by the bi-Lipschitz property of h (Assumption 1), the inverse h^{-1} exists on $h(\mathcal{X})$, so that for any reward function whose composition with h^{-1} lies in the RKHS associated with the RBF kernel, the RFF construction of (Rahimi & Recht, 2007) provides a uniform approximation that becomes arbitrarily accurate as $D_L \to \infty$. The role of SNGP is structural, without the bi-Lipschitz property, h^{-1} is undefined, and even an asymptotic linear representation cannot be motivated.*

Assumption 5 (Matched Truncated Bayesian Prior)**.** *The coefficients $\{\boldsymbol{\beta}^{k*}\}_{k=1}^{K}$ of Assumption 4 are drawn independently across arms from a truncated Gaussian prior:*

$$\boldsymbol{\beta}^{k*} \;\sim\; \pi^k \;:=\; \mathcal{N}\big(\hat{\boldsymbol{\mu}}^k,\, \hat{\boldsymbol{\Sigma}}^k\big)\,\Big|_{\,\|\boldsymbol{\beta}\|\le B}, \qquad k = 1, \dots, K, \tag{13}$$

that is, $\mathcal{N}(\hat{\boldsymbol{\mu}}^k, \hat{\boldsymbol{\Sigma}}^k)$ conditioned on $\{\|\boldsymbol{\beta}\| \le B\}$. The hyperparameters $(\hat{\boldsymbol{\mu}}^k, \hat{\boldsymbol{\Sigma}}^k)$ are the offline-trained VBLL posterior moments. Each $\boldsymbol{\beta}^{k*}$ is drawn once at $t = 0$ and held fixed throughout episodes $t = 1, \dots, T$; the algorithm observes only (s_t, ℓ_t, r_t) but not $\boldsymbol{\beta}^{k*}$ directly.

Remark (Role of truncation)**.** *The truncation in Assumption 5 aligns the support of the prior with the norm bound in Assumption 4, which both algorithms require: the LinUCB regret bound (Lemma 5) holds conditional on every $\boldsymbol{\beta}^{k*}$ in $\{\|\boldsymbol{\beta}\| \le B\}$, and the distribution-independent Bayes risk bound of (Russo & Van Roy, 2014) requires the parameter lie almost surely in a bounded set. In practice, B is set far in the Gaussian tail, so the truncated and untruncated priors are empirically indistinguishable, and the gap with our implementation is negligible.*

Remark (Reduction of the regret decomposition)**.** *Under Assumption 4,*

$$\begin{aligned}\Delta_t &= \max_k \big[f^k_{\text{true}}(\mathbf{x}_t) - c_k\big] - \big[f^{a_t}_{\text{true}}(\mathbf{x}_t) - c_{a_t}\big] \\ &= \max_k \big[\boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}_t - c_k\big] - \big[\boldsymbol{\beta}^{a_t*\top}\boldsymbol{\phi}_t - c_{a_t}\big] \;=\; \Delta_t^\dagger\end{aligned}$$

almost surely, where

$$\Delta_t^\dagger := \max_k \big[\boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}_t - c_k\big] - \big[\boldsymbol{\beta}^{a_t*\top}\boldsymbol{\phi}_t - c_{a_t}\big]. \tag{14}$$

Consequently, the Bayesian regret coincides with $\mathbb{E}[\sum_t \Delta_t^\dagger]$, and the remainder of this appendix bounds this quantity for both algorithms.

Definition 1 (Bayesian Regret)**.** *Under the prior $\pi = \bigotimes_k \pi^k$ of Assumption 5, define*

$$\text{BayesRegret}(T) \;:=\; \mathbb{E}_{\boldsymbol{\beta}^*\sim\pi}\left[\mathbb{E}_{algo}\left[\sum_{t=1}^{T}\Delta_t \,\middle|\, \boldsymbol{\beta}^*\right]\right], \tag{15}$$

where the inner expectation is over the algorithm's internal randomness and the reward noise $\{\xi_t\}_{t=1}^T$, conditional on the realized parameter $\boldsymbol{\beta}^*$.

Table 13: Routing performance under stochastic human availability on CALVIN.

p (human unavailable)	Oracle	BARM (TS)	Oracle − BARM
0.00	0.851	0.776	0.075
0.25	0.821	0.753	0.068
0.50	0.794	0.728	0.066
0.75	0.765	0.682	0.083

Table 14: Selection percentage of two additional execution modes introduced only at deployment on CALVIN.

Arm	0–100	100–200	200–300	300–400	400–500
Extra 4 ($c = 0.45$, reward $= 0.15$)	3.6	1.2	0.4	0.2	0.6
Extra 5 ($c = 0.40$, reward $= 0.25$)	4.0	1.8	0.8	0.4	0.4

G.2 BARM: Bayesian Regret Bound

G.2.1 Reduction to a Linear Contextual Bandit

We let exploration scale $\alpha = 1$ and we reduce the K-arm cost-aware setting to a single d-dimensional linear contextual bandit by concatenating the per-arm parameters. Define

$$\tilde{\theta}^* := \big(\boldsymbol{\beta}^{1*}; \ldots; \boldsymbol{\beta}^{K*}\big) \in \mathbb{R}^d, \qquad d := KD_L, \tag{16}$$

and at each round t the per-arm context vectors

$$\tilde{\psi}_{k,t} := e_k \otimes \boldsymbol{\phi}_t \in \mathbb{R}^d, \qquad k \in \{1, \ldots, K\}, \tag{17}$$

where $e_k \in \mathbb{R}^K$ is the k-th standard basis vector. The cost-adjusted expected reward of arm k at round t is

$$\tilde{\theta}^{*\top} \tilde{\psi}_{k,t} \; - \; c_k \;=\; \boldsymbol{\beta}^{k*\top} \boldsymbol{\phi}_t \; - \; c_k, \tag{18}$$

which is an *affine* function of $\tilde{\theta}^*$ with the known deterministic offset $-c_k$. Such known offsets do not affect the regret analysis. Although $c_{a_t^\dagger} \neq c_{a_t}$ in general, the offsets enter every arm's reward as a deterministic, $\boldsymbol{\beta}^{k*}$-independent shift, so they neither alter the likelihood of r_t given $(\tilde{\theta}^*, a_t, \boldsymbol{\phi}_t)$ nor the algorithm's posterior over $\tilde{\theta}^*$. Equivalently, working with the shifted observation $\tilde{r}_t := r_t - c_{a_t}$ recovers a standard linear contextual bandit with the same sub-Gaussian noise and the same posterior, to which the analysis of (Russo & Van Roy, 2014) applies; the parameter-dependent component of the regret is precisely $\tilde{\theta}^{*\top}(\tilde{\psi}_{a_t^\dagger,t} - \tilde{\psi}_{a_t,t})$.

The arm-selection rule of BARM (Eq. equation 6) is equivalent to the contextual posterior sampling rule $a_t = \arg\max_k(\tilde{\theta}_t^\top \tilde{\psi}_{k,t} - c_k)$, where $\tilde{\theta}_t$ is sampled from the concatenated VBLL posterior. Under Assumption 5, the prior over $\tilde{\theta}^*$ is the truncated Gaussian obtained by stacking the per-arm priors $\{\pi^k\}_{k=1}^K$ as block-independent components, supported on $\{\|\tilde{\theta}\| \le \sqrt{KB^2}\}$.

G.2.2 Linear-Model Regret of TS

Lemma 2 (TS Linear-Model Regret)**.** *Under Assumptions 2, 3, 4, and 5, the BARM algorithm satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta_t^\dagger\right] \;=\; \widetilde{O}\Big(KD_L\sqrt{T}\Big), \tag{19}$$

where the expectation is taken over the joint law of the prior $\pi = \bigotimes_k \pi^k$ and the algorithm's posterior-sampling randomness, and $\widetilde{O}$ hides factors logarithmic in T and constants depending on σ, B, and $c_{\max}$.

Proof. The reduction in Section G.2.1 casts BARM as posterior sampling on a $d = KD_L$-dimensional contextual linear bandit with bounded contexts, bounded parameter, and a matched truncated Gaussian

prior. Russo and Van Roy (Russo & Van Roy, 2014) bound the Bayesian regret of posterior sampling under exactly these conditions; their Proposition 3 gives

$$\mathrm{BayesRisk}(T, \pi^{PS}) = O(d\sqrt{T}\log T) = \widetilde{O}(d\sqrt{T}).$$

Substituting $d = KD_L$ yields equation 19.

□

G.2.3 Main Theorem

Theorem 2 (Bayesian Regret of BARM)**.** *Under Assumptions 1–5,*

$$\mathrm{BayesRegret}(T) \;=\; \widetilde{O}\Big(KD_L\,\sqrt{T}\Big). \tag{20}$$

Proof. Combine Remark G.1 with Lemma 2. □

G.3 SNGP+LinUCB: Bayesian Regret Bound

G.3.1 LinUCB Sufficient Statistics and Arm Selection

For each arm k, LinUCB maintains ridge-regression estimates on the SNGP features $\boldsymbol{\phi}_t \in \mathbb{R}^{D_L}$:

$$\mathbf{V}_t^k = \lambda \mathbf{I}_{D_L} + \sum_{\substack{s<t \\ a_s=k}} \boldsymbol{\phi}_s \boldsymbol{\phi}_s^\top, \tag{21}$$

$$\hat{\boldsymbol{\beta}}_t^k = (\mathbf{V}_t^k)^{-1} \sum_{\substack{s<t \\ a_s=k}} r_s \boldsymbol{\phi}_s. \tag{22}$$

At each episode t, LinUCB selects

$$a_t = \arg\max_{k\in\{1,\dots,K\}} \big[\hat{\boldsymbol{\beta}}_t^{k\top}\boldsymbol{\phi}_t + \beta_T\, w_{k,t} - c_k\big], \qquad w_{k,t} := \sqrt{\boldsymbol{\phi}_t^\top (\mathbf{V}_t^k)^{-1}\boldsymbol{\phi}_t}. \tag{23}$$

G.3.2 Confidence Bound

Lemma 3 (Self-Normalized Confidence Bound)**.** *Under Assumptions 2, 3, and 4, for any fixed $\boldsymbol{\beta}^* = (\boldsymbol{\beta}^{1*}, \dots, \boldsymbol{\beta}^{K*})$ with $\|\boldsymbol{\beta}^{k*}\| \le B$, define*

$$\beta_T := \sigma \sqrt{D_L\,\log\Big(1 + \frac{T\,C_\phi^2}{D_L\,\lambda}\Big) + 2\log\frac{K}{\delta}} + \sqrt{\lambda}\,B. \tag{24}$$

Then with probability $\ge 1-\delta$ over the algorithm's randomness (conditional on $\boldsymbol{\beta}^*$), simultaneously for all $t \le T$ and all $k \in \{1,\dots,K\}$:

$$\big|\boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}_t - \hat{\boldsymbol{\beta}}_t^{k\top}\boldsymbol{\phi}_t\big| \le \beta_T\, w_{k,t}. \tag{25}$$

Proof. **Step 1: Single-arm self-normalized bound.** Define the noise sum $\mathbf{M}_t^k := \sum_{s<t,\, a_s=k} \xi_s\, \boldsymbol{\phi}_s$, where $\xi_s := r_s - \boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}_s$ is the realized regression noise. Substituting $r_s = \boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}_s + \xi_s$ into equation 22 gives equation 26. Substituting r_s into equation 22

$$\hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}^{k*} = (\mathbf{V}_t^k)^{-1}\mathbf{M}_t^k - \lambda\,(\mathbf{V}_t^k)^{-1}\boldsymbol{\beta}^{k*}. \tag{26}$$

Applying the generalized Cauchy–Schwarz inequality $|u^\top v| \le \|u\|_M \|v\|_{M^{-1}}$ with $M = \mathbf{V}_t^k$, $u = \hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}^{k*}$, and $v = \boldsymbol{\phi}_t$ gives

$$\big|\boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}_t - \hat{\boldsymbol{\beta}}_t^{k\top}\boldsymbol{\phi}_t\big| \le \|\hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}^{k*}\|_{\mathbf{V}_t^k} \cdot w_{k,t}. \tag{27}$$

Applying the triangle inequality in the $(\mathbf{V}_t^k)^{-1}$-norm to equation 26,

$$\|\hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}^{k*}\|_{\mathbf{V}_t^k} \le \|\mathbf{M}_t^k\|_{(\mathbf{V}_t^k)^{-1}} + \lambda\,\|\boldsymbol{\beta}^{k*}\|_{(\mathbf{V}_t^k)^{-1}}. \tag{28}$$

For the noise term, Theorem 1 of (Abbasi-Yadkori et al., 2011) gives, with probability $\geq 1-\delta'$ uniformly for all $t \leq T$,

$$\|\mathbf{M}_t^k\|_{(\mathbf{V}_t^k)^{-1}} \leq \sigma \sqrt{2\log\frac{1}{\delta'} + \log\frac{\det(\mathbf{V}_t^k)}{\det(\lambda\mathbf{I})}}. \tag{29}$$

For the regularization term, $\mathbf{V}_t^k \succeq \lambda\mathbf{I}$ implies $(\mathbf{V}_t^k)^{-1} \preceq \lambda^{-1}\mathbf{I}$, so

$$\lambda\,\|\boldsymbol{\beta}^{k*}\|_{(\mathbf{V}_t^k)^{-1}} \leq \sqrt{\lambda}\,\|\boldsymbol{\beta}^{k*}\| \leq \sqrt{\lambda}\,B, \tag{30}$$

where the second inequality uses $\|\boldsymbol{\beta}^{k*}\| \leq B$ from Assumption 4.

Step 2: Log-determinant bound. By AM–GM applied to the eigenvalues of $\mathbf{V}_t^k$ together with $\mathrm{tr}(\mathbf{V}_t^k) \leq D_L\lambda + TC_\phi^2$ (using Assumption 3 and $|\{s<t : a_s = k\}| \leq T$),

$$\log\frac{\det(\mathbf{V}_t^k)}{\det(\lambda\mathbf{I})} \leq D_L \log\Big(1 + \frac{TC_\phi^2}{D_L\lambda}\Big). \tag{31}$$

Step 3: Union bound over K arms. Combining equation 27–equation 31, for a single arm k with probability $\geq 1-\delta'$,

$$\big|\boldsymbol{\beta}^{k*\top}\boldsymbol{\phi}_t - \hat{\boldsymbol{\beta}}_t^{k\top}\boldsymbol{\phi}_t\big| \leq \left[\sigma\sqrt{D_L\log\Big(1+\frac{TC_\phi^2}{D_L\lambda}\Big) + 2\log\frac{1}{\delta'}} + \sqrt{\lambda}B\right] w_{k,t}. \tag{32}$$

Setting $\delta' = \delta/K$ and taking a union bound over $k = 1, \ldots, K$, the failure probability is at most δ. Substituting $2\log(1/\delta') = 2\log(K/\delta)$ yields equation 25 with β_T as in equation 24. □

G.3.3 Elliptic Potential Lemma

Lemma 4 (Elliptic Potential). *For any sequence $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_T \in \mathbb{R}^{D_L}$ with $\|\boldsymbol{\phi}_t\| \leq C_\phi$, and any arm k with $n_k := |\{t \leq T : a_t = k\}|$ selections, define $C_\lambda := 1 + C_\phi^2/\lambda$. Then*

$$\sum_{\substack{t=1\\a_t=k}}^{T} \boldsymbol{\phi}_t^\top (\mathbf{V}_t^k)^{-1}\boldsymbol{\phi}_t \leq C_\lambda D_L \log\Big(1 + \frac{n_k C_\phi^2}{D_L\lambda}\Big). \tag{33}$$

Proof. This is (Abbasi-Yadkori et al., 2011, Lemma 11) specialized to the samples assigned to arm k with feature norm bound C_ϕ and regularization λ. Letting $s_t := \boldsymbol{\phi}_t^\top(\mathbf{V}_t^k)^{-1}\boldsymbol{\phi}_t$, the matrix determinant lemma gives the telescoping identity $\sum_{a_t=k}\log(1+s_t) = \log\det(\mathbf{V}_{T+1}^k)/\det(\lambda\mathbf{I})$. For $s \in [0, C_\phi^2/\lambda]$, the integral representation of $\log$ yields $s \leq C_\lambda\log(1+s)$, hence $\sum_{a_t=k} s_t \leq C_\lambda \log\det(\mathbf{V}_{T+1}^k)/\det(\lambda\mathbf{I})$. Bounding the log-determinant by the AM–GM trace argument as in equation 31 (with T replaced by n_k) yields equation 33. □

G.3.4 LinUCB Conditional Frequentist Regret

Lemma 5 (Conditional Frequentist Regret of LinUCB). *Under Assumptions 2, 3, and 4, with $\delta = 1/T$, for any fixed $\boldsymbol{\beta}^* = (\boldsymbol{\beta}^{1*}, \ldots, \boldsymbol{\beta}^{K*})$ satisfying $\|\boldsymbol{\beta}^{k*}\| \leq B$,*

$$\mathbb{E}_{\mathrm{algo}}\left[\sum_{t=1}^{T}\Delta_t^\dagger \,\Big|\, \boldsymbol{\beta}^*\right] \leq 2\beta_T\sqrt{C_\lambda K D_L T\log\Big(1+\frac{TC_\phi^2}{KD_L\lambda}\Big)} + B_{\max} \;=:\; g(T), \tag{34}$$

where $B_{\max} = 1 + c_{\max}$, β_T is equation 24 with $\delta = 1/T$, and $g(T)$ is independent of $\boldsymbol{\beta}^*$ given the norm bound.

Proof. Set $\delta = 1/T$ and let $\mathcal{E}$ denote the confidence event of Lemma 3, so $\mathbb{P}(\mathcal{E} \mid \boldsymbol{\beta}^*) \geq 1 - 1/T$.

Step 1: Per-step regret under $\mathcal{E}$. Under $\mathcal{E}$, applying equation 25 as a lower confidence bound on the linear-model optimal arm $a_t^\dagger$ and as an upper confidence bound on the selected arm a_t,

$$\begin{aligned}\boldsymbol{\beta}^{a_t^\dagger *\top}\boldsymbol{\phi}_t - c_{a_t^\dagger} &\le \hat{\boldsymbol{\beta}}_t^{a_t^\dagger\top}\boldsymbol{\phi}_t + \beta_T w_{a_t^\dagger,t} - c_{a_t^\dagger}\\ &\le \hat{\boldsymbol{\beta}}_t^{a_t\top}\boldsymbol{\phi}_t + \beta_T w_{a_t,t} - c_{a_t}\\ &\le \boldsymbol{\beta}^{a_t *\top}\boldsymbol{\phi}_t + 2\beta_T w_{a_t,t} - c_{a_t},\end{aligned}$$

where the second inequality uses the UCB optimality of a_t under rule equation 23. Rearranging,

$$\Delta_t^\dagger \le 2\beta_T w_{a_t,t}. \tag{35}$$

Step 2: Cauchy–Schwarz. Summing equation 35 over t,

$$\sum_{t=1}^T \Delta_t^\dagger \le 2\beta_T \sqrt{T\sum_{t=1}^T w_{a_t,t}^2}. \tag{36}$$

Step 3: Elliptic potential across K arms. Splitting by arm and applying Lemma 4,

$$\sum_{t=1}^T w_{a_t,t}^2 = \sum_{k=1}^K \sum_{t:a_t=k} \boldsymbol{\phi}_t^\top (\mathbf{V}_t^k)^{-1}\boldsymbol{\phi}_t \le C_\lambda D_L \sum_{k=1}^K \log\Big(1+\frac{n_k C_\phi^2}{D_L\lambda}\Big).$$

Since $f(x)=\log(1+xC_\phi^2/(D_L\lambda))$ is concave and $\sum_k n_k = T$, Jensen's inequality with uniform weights gives

$$\sum_{t=1}^T w_{a_t,t}^2 \le C_\lambda K D_L \log\Big(1+\frac{TC_\phi^2}{KD_L\lambda}\Big) =: \Gamma_T^{\mathrm{UCB}}. \tag{37}$$

Step 4: Combine with failure event. Under $\mathcal{E}$, $\sum_t \Delta_t^\dagger \le 2\beta_T\sqrt{T\Gamma_T^{\mathrm{UCB}}}$. The per-step regret is bounded almost surely by $B_{\max} = 1 + c_{\max}$ since $r_t \in [0,1]$ and $c_k \in [0, c_{\max}]$. By the law of total expectation conditional on $\boldsymbol{\beta}^*$,

$$\mathbb{E}_{\mathrm{algo}}\left[\sum_{t=1}^T \Delta_t^\dagger \,\Big|\, \boldsymbol{\beta}^*\right] \le 2\beta_T\sqrt{T\Gamma_T^{\mathrm{UCB}}}\cdot\mathbb{P}(\mathcal{E}\mid\boldsymbol{\beta}^*) + B_{\max}\,T\cdot\mathbb{P}(\mathcal{E}^c\mid\boldsymbol{\beta}^*) \le 2\beta_T\sqrt{T\Gamma_T^{\mathrm{UCB}}} + B_{\max}.$$

The right-hand side depends on $\boldsymbol{\beta}^*$ only through the deterministic norm bound $\|\boldsymbol{\beta}^{k*}\| \le B$ used in equation 30, and is otherwise free of $\boldsymbol{\beta}^*$. □

G.3.5 Main Theorem

Theorem 3 (Bayesian Regret of SNGP+LinUCB)**.** *Under Assumptions 1–5,*

$$\mathrm{BayesRegret}(T) \le 2\beta_T\sqrt{C_\lambda K D_L T\log\Big(1+\frac{TC_\phi^2}{KD_L\lambda}\Big)} + B_{\max} \;=\; \widetilde{O}\big(D_L\sqrt{KT}\big), \tag{38}$$

where $C_\lambda = 1 + C_\phi^2/\lambda$, $B_{\max} = 1 + c_{\max}$, and β_T is equation 24 with $\delta = 1/T$.

Proof. By Lemma 5, for every fixed $\boldsymbol{\beta}^* = (\boldsymbol{\beta}^{1*},\dots,\boldsymbol{\beta}^{K*})$ in $\{\|\boldsymbol{\beta}^{k*}\| \le B,\ \forall k\}$,

$$\mathbb{E}_{\mathrm{algo}}\left[\sum_{t=1}^T \Delta_t^\dagger \,\Big|\, \boldsymbol{\beta}^*\right] \le g(T), \tag{39}$$

where $g(T)$ is the right-hand side of equation 38 and is independent of $\boldsymbol{\beta}^*$. By Assumption 5, the prior $\pi = \bigotimes_k \pi^k$ is supported entirely within $\{\|\boldsymbol{\beta}^{k*}\| \le B,\ \forall k\}$. Taking expectation of equation 39 over $\boldsymbol{\beta}^* \sim \pi$ and combining with Remark G.1,

$$\begin{aligned}\mathrm{BayesRegret}(T) &= \mathbb{E}_{\boldsymbol{\beta}^*\sim\pi}\left[\mathbb{E}_{\mathrm{algo}}\left[\sum_{t=1}^T \Delta_t^\dagger \,\Big|\, \boldsymbol{\beta}^*\right]\right]\\ &\le \mathbb{E}_{\boldsymbol{\beta}^*\sim\pi}[g(T)] \;=\; g(T),\end{aligned}$$

since $g(T)$ is a deterministic constant. The asymptotic form $\widetilde{O}(D_L\sqrt{KT})$ follows by expanding β_T and Γ_T^{UCB} and collecting polylogarithmic factors. □

G.4 Comparison of TS and LinUCB Regret Bounds

Under the unified Bayesian regret framework established by Assumptions 4–5, both Theorem 2 and Theorem 3 bound the *same* quantity, $\mathrm{BayesRegret}(T)$, on the *same* cost-adjusted contextual bandit. Their leading terms are

$$\text{TS leading term} \;=\; \widetilde{O}\big(KD_L\sqrt{T}\big), \tag{40}$$

$$\text{LinUCB leading term} \;=\; \widetilde{O}\big(D_L\sqrt{KT}\big), \tag{41}$$

treating σ, λ, B, $c_{\max}$ as constants and absorbing polylogarithmic factors into $\widetilde{O}$. Both bounds are $\widetilde{O}(\sqrt{T})$ in the horizon and polynomial in (K, D_L). Their ratio is

$$\frac{\text{TS bound}}{\text{LinUCB bound}} \;=\; \widetilde{O}\big(\sqrt{K}\big), \tag{42}$$

indicating that the worst-case TS *upper bound* we obtain is *looser* than the LinUCB upper bound by a factor of $\widetilde{O}(\sqrt{K})$.

Remark (Position relative to the TS literature)**.** *The $\sqrt{K}$ factor by which our TS bound exceeds the LinUCB bound mirrors the $\sqrt{d}$ gap between TS (Agrawal & Goyal, 2013) and LinUCB (Abbasi-Yadkori et al., 2011) for contextual linear bandits, and distribution-independent Bayes risk analysis (Russo & Van Roy, 2014) does not exploit the K-armed structure. Tighter information-ratio analysis (Russo & Van Roy, 2016) requires fixed action sets, whereas our reduction in Section G.2.1 yields a stochastic time-varying action set $\{\tilde{\psi}_{k,t}\}_{k=1}^{K}$ through the context ϕ_t; extending information-ratio analysis to context-dependent action sets remain an open technical problem. TS nonetheless beats UCB empirically, both here and in earlier studies (Chapelle & Li, 2011). The worst-case analysis we use here cannot see this gap, because it averages over priors that the algorithm in fact exploits.*

G.5 Theory–Ablation Alignment

Theorem 2 relies on three structural conditions: that the hidden mapping h be bi-Lipschitz (Assumption 1), that the prior on $\boldsymbol{\beta}^{k*}$ be Gaussian with bounded support (Assumption 5), and that the arm be drawn from this posterior at every round. Each of the three components in BARM corresponds to one of these conditions. Spectral normalization is the mechanism that makes h bi-Lipschitz; without it, h^{-1} on $h(\mathcal{X})$ is undefined and the linear-realizability statement of Assumption 4 loses its motivation. The Gaussian posterior comes from VBLL training, with the norm bound $\|\boldsymbol{\beta}\| \le B$ absorbed into the truncation. Posterior sampling is what TS does by construction.

The ablations in Section 5 can be read against this correspondence. Replacing SNGP with a plain MLP breaks bi-Lipschitzness in the most direct way visible in our setup: Figure 7(c) shows the singular values of the MLP feature matrix collapsing to $\sim 10^{-11}$ past the 15th dimension, which is exactly the failure mode that injectivity of h rules out. Replacing VBLL with MC Dropout retains a notion of posterior, but not the Gaussian form that Assumption 5 requires; Table 3 reports an offline-OOD NLL of 9.04 for MCDropout against -1.10 for BARM, an order-of-magnitude gap on the very inputs where calibration matters. Replacing TS with greedy selection keeps both the features and the posterior, and discards only the sampling step; the corresponding drop on CALVIN is the smallest of the three (-6.2pp), as one would expect from removing only the last of the three conditions.

We do not claim the regret bound predicts these gaps quantitatively. The point is narrower: the direction of each ablation drop matches that of which condition it removes, and the relative order of the drops (SNGP $>$ VBLL $>$ TS in damage caused) is the order in which the conditions enter the proof.