ACTIVATIONSMC: SEQUENTIAL MONTE CARLO FOR OPTIMIZING ACTIVATION STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Activation steering controls pretrained language models and beyond at inference by modifying internal activations, but its effectiveness hinges on jointly deciding where and how strongly to intervene. This creates a hard optimization over a mixed discrete-continuous space with coupled layer choices and often non-differentiable rewards, poorly handled by fixed grids or greedy decoupled search. We introduce **ActivationSMC**, a derivative-free black-box framework based on Sequential Monte Carlo that maintains a population of full steering configurations. It guides the population toward high-reward regions via reweighting, resampling, and stochastic mutation that preserves diverse modes, with a multi-fidelity schedule that uses small subsets for cheap early exploration and larger subsets for precise refinement. Without updating model weights, ActivationSMC configures off-the-shelf activation steering operators such as COLD-Steer, AUSteer, CAA, and Spherical Steering. Across TruthfulQA, R-Bench, AMBER, as well as text-to-image generation using Stable Diffusion, ActivationSMC consistently outperforms native tuning. It improves collapsed baselines by up to 28.9 percentage points, raises compositional text-to-image alignment by 20–30 percentage points across four steering operators, and substantially widens its lead as joint multi-layer optimization exposes the breakdown of decoupled heuristics.

1 INTRODUCTION

Activation steering is an inference-time substrate for controlling pretrained transformers and beyond. By modifying activations during the forward pass, it enables post-training alignment, safety, and style adaptation without weight updates. The paradigm has progressed rapidly from adding contrastive vectors to the residual stream (Turner et al., 2023; Rimsky et al., 2024; Zou et al., 2023) to learned, sparse, or geometric transformations of internal states (Wu et al., 2024b; Sharma & Trivedi, 2026; Feng et al., 2026; Vu & Nguyen, 2025; You et al., 2026). Parallel work has refined *where* to intervene via precise localization across layers, heads, and units (Li et al., 2023; Lai et al., 2025; Gadgil et al., 2026; Nelwan & Wicaksono, 2026), and *how* via dynamic, input-dependent scaling (Lee et al., 2025; Hsu et al., 2026; Ferrando et al., 2025; Parekh et al., 2025).

Despite remarkable progress in activation steering, its full potential hinges on a joint decision that current practice treats as separable: where to steer and how strongly. Early interventions alter downstream activations, so co-dependent layers cannot be optimized independently. The resulting objective is mixed discrete-continuous, non-convex, non-separable, and often non-differentiable, with irregular, multi-modal reward surfaces (Feng et al., 2026; You et al., 2026; Robertson et al., 2026; Bas & Novak, 2026). As more layers are jointly steered, cross-layer coupling dominates.

Existing tuning strategies for steering configurations, which layers to steer and activation steering operator-related hyperparameters, are ill-suited to this joint space. Most rely on restrictive heuristics: small fixed grids over layer-strength pairs in COLD-Steer (Sharma & Trivedi, 2026), one-dimensional sweeps over support size in AUSteer (Feng et al., 2026), or greedy decoupled selection in Spherical Steering (You et al., 2026). More advanced alternatives use differentiable optimization for joint layer-coefficient learning (Lai et al., 2025) or geometry-guided Bayesian optimization (Robertson et al., 2026), but require differentiable rewards or remain limited to single layers. The optimizer is thus treated as an implementation detail rather than part of the steering design.

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

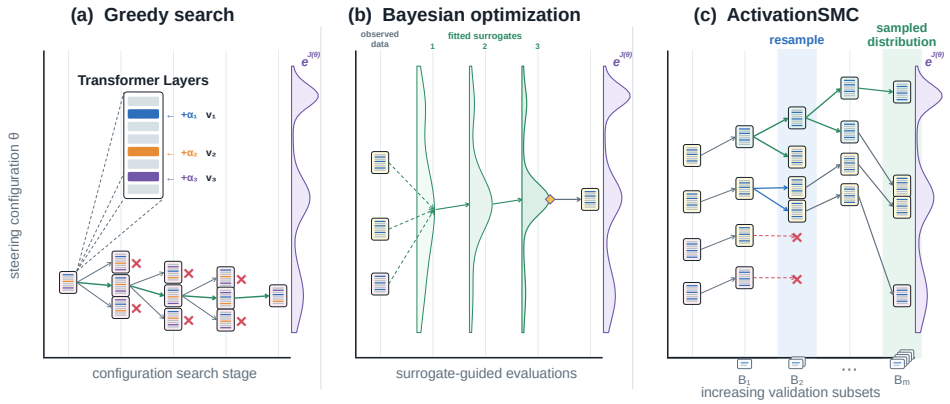


Figure 1: **ActivationSMC vs. existing search.** Greedy search decouples layers and Bayesian optimization struggles to find high-reward modes. ActivationSMC maintains a population of multi-layer configurations, preserves competing modes through reweighting and resampling, and gradually sharpens reward estimates via a multi-fidelity schedule.

1.1 Our Contributions

To address the bottlenecks of fixed grids and decoupled heuristics in the joint, non-differentiable optimization of where and how strongly to intervene, we introduce **ActivationSMC**, a derivative-free black-box framework that directly maximizes task rewards leveraging sequential Monte Carlo (SMC) (Naesseth et al., 2019). Figure 1 compares ActivationSMC with other existing methods. Our particular contributions are three-fold:

1. Joint SMC over the hybrid configuration space. We formulate optimizing steering configurations as sampling from reward-tilted Gibbs distributions. ActivationSMC maintains a population of configurations evolved via reweighting, resampling, and mutation with local and global proposals.

2. Data-graduated multi-fidelity search with effective sample size (ESS)-controlled tempering. We introduce nested subsets $\mathcal{B}_1 \subset \cdots \subset \mathcal{B}_M$ across stages with ESS-controlled adaptive tempering within each stage. Cheap noisy estimates drive broad early exploration while larger subsets refine estimates as the population concentrates, greatly reducing cost without loss of precision.

3. Extensive validations. We optimize four prominent steering operators—CAA (Rimsky et al., 2024), COLD-Steer (Sharma & Trivedi, 2026), AUSteer (Feng et al., 2026), and Spherical Steering (You et al., 2026)—across text-to-text (TruthfulQA), image-to-text (R-Bench, AMBER), and text-to-image (SD3.5 Large Turbo). ActivationSMC outperforms native tuning, rescues failing baselines, and its lead widens from single to multi-layer steering where coupling is strongest.

1.2 Additional Related Works

Optimizing steering configurations. JoLA (Lai et al., 2025) jointly learns layers and hyperparameters but requires differentiable training and backpropagation through the model. GRACE (Robertson et al., 2026) is black-box via geometry-guided Bayesian optimization, but searches only a single layer and a single coefficient. ActivationSMC instead performs black-box search over multi-layer, mixed discrete-continuous configurations with layer-specific parameters, using only sequence-level, possibly non-differentiable rewards.

Sampling-based steering of language models. A parallel line uses SMC to control generation itself rather than the intervention that enables it. SMC steering formulates constrained generation as posterior inference over token sequences (Lew et al., 2023), extended to programmable syntactic and semantic constraints (Loula et al., 2025). Twisted SMC (Zhao et al., 2024) learns intermediate potentials to estimate future value of partial sequences, and recent variants apply particle filtering to reasoning rollouts or use a planner to construct inference programs for a follower population (Puri et al., 2025; Grand et al., 2025). These methods place particles over tokens or reasoning trajectories and score them during generation, requiring learned twists to avoid degeneracy. ActivationSMC inverts this: particles are over configurations of steering operators, scored from complete model

outputs. This different particle semantics is what enables its multi-fidelity schedule with nested subsets and ESS-controlled tempering.

1.3 Organization

The remainder of the paper is organized as follows. Section 2 formalizes the steering problem and reviews SMC, Section 3 introduces ActivationSMC and its multi-fidelity ESS-controlled search, Section 4 presents experimental results across three modalities, and Section 5 concludes. Further methodological and experimental details are provided in the appendix.

2 Background

2.1 The Steering Problem

Activation steering modifies a pretrained transformer by intervening at internal sites, e.g., layers or heads, during the forward pass. For a fixed steering operator, a configuration $\theta \in \Theta$ specifies both *where* to intervene and *how*. Concretely, $\theta = (\mathcal{L}, \theta_{1:K})$, where $\mathcal{L} = (l_1, \dots, l_K)$ with $1 \le l_1 < \cdots < l_K \le L$ specifies K intervention layers, and θ_j contains the local steering parameters at layer l_j. These parameters may be continuous, integer, or categorical. The exact parametrization per operator is in Appendix D.

For the chosen steering operator, we aim to maximize expected task reward under the steered model:

$$\theta^\star \in \arg\max_{\theta \in \Theta} J(\theta), \qquad J(\theta) = \mathbb{E}_{x \sim p_{\mathrm{data}},\, y \sim p_\theta(\cdot|x)}[r(x, y)], \tag{1}$$

where y is the model output for input x, p_{data} is the task input distribution, $p_\theta(y \mid x)$ is the conditional output distribution under steering policy θ, and $r(x, y)$ is a sequence-level, possibly non-differentiable reward, e.g., binary correctness or compositional alignment. In practice we access J only via a noisy empirical estimate $\widehat{J}(\theta; \mathcal{B})$ on a finite validation set $\mathcal{B}$.

Notice that the objective function J is mixed discrete-continuous, non-convex, and non-separable: early interventions alter downstream activations, coupling parameters across layers and inducing irregular response surfaces. To enable population-based optimization that explores competing modes, we target its Gibbs smoothing

$$\pi_T(\theta) \propto \exp\left(\frac{J(\theta)}{T}\right) \nu_0(\theta), \tag{2}$$

where ν_0 is a reference density over Θ (e.g., uniform over layers and bounded strengths) and $T > 0$ is a temperature. For $T > 0$, π_T can retain multiple promising configurations; as $T \to 0$, it concentrates on maximizers of J under suitable support and regularity assumptions.

2.2 Sequential Monte Carlo

Sampling directly from the Gibbs target (2) is intractable when T is small. SMC addresses this by bridging ν_0 to π_T through a sequence of intermediate distributions. For a fixed reward J, a canonical construction uses a tempering schedule $0 = \tau_0 < \tau_1 < \cdots < \tau_M = 1/T$ with

$$\pi_m(\theta) = \frac{\exp(\tau_m J(\theta))}{Z_m} \nu_0(\theta), \qquad G_m(\theta) = \exp\big((\tau_m - \tau_{m-1}) J(\theta)\big),$$

where Z_m is the normalizing constant. More generally, π_m for $m < M$ can be designed for computational advantage, e.g., via cheap approximations of J as in Section 3.3.

SMC initializes N particles from ν_0 and sequentially approximates $\pi_1, \dots, \pi_M$ by iterating reweight, resample, and mutate. Given equally weighted particles $\{\theta_{m-1}^{(i)}\}_{i=1}^N$ approximating π_{m-1}, the selection step assigns

$$w_i = \frac{G_m(\theta_{m-1}^{(i)})}{\sum_{j=1}^N G_m(\theta_{m-1}^{(j)})}, \qquad (A_1, \dots, A_N) \sim \mathrm{Resample}(w_{1:N}),$$

and sets $\widetilde{\theta}^{(i)} = \theta_{m-1}^{(A_i)}$, e.g., via systematic resampling. Thus high-reward configurations are retained and replicated, which can reduce diversity.

Method	θ_j	Meaning
CAA (Rimsky et al., 2024)	α_j	Scale of contrastive direction
COLD-Steer (Sharma & Trivedi, 2026)	α_j	Scale of learned update
AUSteer (Feng et al., 2026)	$(\log \alpha_j, k_j)$	Rescaling strength and number of edited units
Spherical Steering (You et al., 2026)	(α_j, β_j)	Rotation strength and gating threshold

Table 1: Local search parameters per steering method.

To restore sample diversity, each resampled particle $\widetilde{\theta}^{(i)}$ is mutated by proposing $\theta' \sim q(\cdot \mid \widetilde{\theta}^{(i)})$ and accepting it with probability

$$\min\left\{1, \frac{\pi_m(\theta')\, q(\widetilde{\theta}^{(i)} \mid \theta')}{\pi_m(\widetilde{\theta}^{(i)})\, q(\theta' \mid \widetilde{\theta}^{(i)})}\right\}. \tag{3}$$

The proposal q may change discrete locations or perturb continuous, integer, and categorical hyperparameters, and (3) ensures the mutation leaves π_m invariant. We review convergence guarantees of SMC in Appendix F.

3 ACTIVATIONSMC

Optimizing J in (1) is challenging due to non-convexity, non-separability, mixed discrete-continuous domain, and noisy black-box rewards. We target its Gibbs smoothing (2) with a population-based SMC sampler. The choice is dictated by problem structure: a configuration has $K(p+1)$ variables—K intervention sites and p per-site hyperparameters, typically < 20 for $K \leq 2$—precisely where SMC is provably efficient (Naesseth et al., 2019; Beskos et al., 2016). Maintaining a population of full configurations, it preserves multiple modes via reweighting, resampling, and Metropolis–Hastings (MH) mutation over discrete and continuous coordinates, yields a compact reusable policy (configuration), and naturally enables multi-fidelity, ESS-controlled tempering.

3.1 LAYERWISE STEERING POLICIES

We define a steering policy $\theta = (\mathcal{L}, \theta_{1:K}) \in \Theta$, where $\mathcal{L} = (l_1, \ldots, l_K)$ with $1 \leq l_1 < \cdots < l_K \leq L$ specifies K layers of an L-layer model and θ_j is the local intervention at layer l_j. The operator family and K are fixed within each search.

We consider additive and rotational interventions. At selected layer l_j,

$$h'_{l_j} = \mathcal{T}(h_{l_j}; \theta_j), \qquad \mathcal{T}(h; \theta_j) = \begin{cases} h + \Delta(h; \theta_j), & \text{additive,} \\ Q(h; \theta_j)h, & \text{rotational,} \end{cases} \tag{4}$$

where Δ is an additive update and $Q^\top Q = I$ is a rotation. Both may depend on the activation and on fixed steering signals for that layer. Table 1 summarizes the local parameters for CAA, COLD-Steer, AUSteer, and Spherical Steering; full definitions are provided in Appendix D.

3.2 GENERAL REWARDS AND THEIR EVALUATION

Let $p_\theta(y \mid x)$ be the output distribution under policy θ. For task distribution p_{data} and scalar reward r, we maximize

$$J(\theta) = \mathbb{E}_{x \sim p_{\text{data}},\, y \sim p_\theta(\cdot|x)}[r(x, y)]. \tag{5}$$

The reward function $r(x, y)$ may measure truthfulness (Lin et al., 2022), image–text alignment (Hu et al., 2023), or human preference (Ma et al., 2025); only scalar evaluations are required, allowing non-differentiable rewards.

Given a validation subset $\mathcal{B}$, we sample S complete outputs $y_x^{(a)} \sim p_\theta(\cdot \mid x)$, $a = 1, \ldots, S$, and estimate $J(\theta)$ as

$$\widehat{J}(\theta; \mathcal{B}) = \frac{1}{|\mathcal{B}|S} \sum_{x \in \mathcal{B}} \sum_{a=1}^{S} r(x, y_x^{(a)}). \tag{6}$$

In particular, $S = 1$ is single-output evaluation. All policies are evaluated on the same $\mathcal{B}$ with fixed generation seeds; when multiple metrics are combined, each is averaged over $\mathcal{B}$ and replicates before combination. An optional formulation with a preservation penalty is in Appendix C.1.

3.3 MULTI-FIDELITY SEQUENTIAL MONTE CARLO OPTIMIZATION

To concentrate search on promising steering policies, we construct a sequence of reward-tilted distributions and track them with an SMC sampler (Del Moral et al., 2006). Each particle encodes a complete steering configuration. Evaluating every particle at full fidelity is wasteful while the population is diffuse, so we adopt a data-graduated multi-fidelity schedule: small subsets enable cheap, noisy exploration early, while progressively larger subsets refine reward estimates later. This follows the cost–accuracy principle of multi-fidelity and multilevel Monte Carlo (Giles, 2015; Peherstorfer et al., 2016), but fidelity is controlled solely by the cardinality of the evaluation set, with the underlying model and per-output generation procedure held fixed. The search proceeds over M stages with nested subsets

$$\mathcal{B}_1 \subseteq \cdots \subseteq \mathcal{B}_M = \mathcal{D}_{\text{search}},$$

where $\mathcal{D}_{\text{search}}$ is the prompt set reserved for policy optimization. We denote $\widehat{J}_m(\theta) = \widehat{J}(\theta; \mathcal{B}_m)$ for the estimator in (6) with S_m replicates per input.

Target distributions. Let ν_0 be a reference measure on space Θ from which N particles $\theta^{(1)}, \ldots, \theta^{(N)} \overset{\text{i.i.d.}}{\sim} \nu_0$ are drawn. Following the SMC sampler construction (Del Moral et al., 2006), we define stage targets

$$\pi_m(\theta) \propto \nu_0(\theta) \exp\Big(\tau_m \widehat{J}_m(\theta)\Big), \qquad \pi_0 = \nu_0, \tag{7}$$

with prescribed non-decreasing inverse temperatures $0 < \tau_1 \leq \cdots \leq \tau_M$. The exponential tilt reweights ν_0 toward higher empirical reward, and larger τ_m progressively sharpens this preference.

Incremental reweighting. We carry a single population of N complete policies through all M stages. Within stage m, we interpolate from π_{m-1} to π_m via a geometric bridge controlled by a single shared scalar $s \in [0, 1]$,

$$\pi_{m,s}(\theta) \propto \pi_{m-1}(\theta)^{1-s} \pi_m(\theta)^s, \qquad s \in [0, 1]. \tag{8}$$

For particles equally weighted under $\pi_{m,s}$, an advance $s \to s' > s$ assigns

$$w_i \propto \frac{\pi_{m,s'}(\theta^{(i)})}{\pi_{m,s}(\theta^{(i)})}, \qquad \sum_{i=1}^{N} w_i = 1, \tag{9}$$

where normalizing constants cancel upon normalization. To avoid weight degeneracy when $\widehat{J}_m$ or τ_m changes, we choose the largest $s' \in (s, 1]$ such that the effective sample size (ESS) satisfies

$$\text{ESS} = \frac{1}{\sum_{i=1}^{N} w_i^2} \geq \rho N, \qquad 0 < \rho < 1, \tag{10}$$

with ρ controlling allowable concentration. If feasible at $s' = 1$, the stage completes in one step; otherwise s' is found by bisection (Beskos et al., 2016). The inner bridge does not introduce a second layer of SMC: it updates the same particle population with a common s, using the same proposal family $q(\cdot \mid \theta)$ throughout the stage—only the target in the MH acceptance changes with $\pi_{m,s'}$. After each advance we resample and apply the MH updates below.

Resampling and mutation. We systematically resample N particles according to their weights in (9), so particle $\theta^{(i)}$ receives Nw_i copies in expectation. After resampling, all particles receive equal weight $1/N$. We then apply MH updates (Hastings, 1970). Each update selects a local or global proposal component with fixed probabilities. Writing q for the selected component, we propose $\theta' \sim q(\cdot \mid \theta)$ and accept it with probability

$$\min\left\{1, \frac{\pi_{m,s'}(\theta') q(\theta \mid \theta')}{\pi_{m,s'}(\theta) q(\theta' \mid \theta)}\right\}. \tag{11}$$

Thus, the overall mutation kernel combines local and global proposal components. Local proposals perturb intervention layers and their parameters; global proposals redraw a feasible configuration from ν_0. These updates use scalar reward evaluations without requiring derivatives[1]. Infeasible or rejected proposals leave θ unchanged. The MH kernel preserves the current target $\pi_{m,s'}$. We set

Algorithm 1 ActivationSMC: Data-Graduated Multi-Fidelity SMC over Steering Policies

Require: Reference ν_0, population N, nested subsets $\mathcal{B}_{1:M}$, inverse temps $\tau_{1:M}$, ESS fraction ρ, proposals q_m, MH steps K_{MH}
Ensure: Optimized policy $\theta^\star$

```
Sample {θ_i}_{i=1}^N ~ ν_0 i.i.d.; H ← ∅
for m = 1, ..., M do
    s ← 0
    while s < 1 do
        Retrieve/cache Ĵ(θ_i; B_m) for current particles
        Define w_i(s') ∝ exp((s' − s)Δ_m(θ_i))
        Choose largest s' ∈ (s, 1] with ESS(w_{1:N}(s')) ≥ ρN        ▷ (10)
        Compute w_i ← w_i(s') via (9)
        H ← H ∪ {θ_i}_{i=1}^N
        Systematically resample {θ_i} by w_{1:N}
        w_i ← 1/N, i = 1, ..., N
        for k = 1, ..., K_MH do
            Propose θ'_i ~ q_m(θ_i, ·) for each i
            Accept via (11) targeting π_{m,s'}
            H ← H ∪ {evaluated proposals}
        end for
        s ← s'
    end while
end for
Re-evaluate shortlist from H ∪ {θ_i} on D_val with common seeds
A ← finalists satisfying preservation constraint
return θ* = arg max_{θ∈A} Ĵ(θ; D_val) or infeasible if A = ∅
```

$s \leftarrow s'$ and repeat reweighting, resampling, and MH updates until $s = 1$, then proceed to the next evaluation stage with $s = 0$.

Final selection. The terminal population approximates π_M. Theorem C.1 in Appendix C gives an optimization-error bound for selection under the final search reward, decomposing the error into reward-estimation, target concentration, and particle approximation. For deployment we remove multi-fidelity bias by re-evaluating on held-out data: we form a candidate pool $\mathcal{A}$ consisting of (i) a shortlist from the terminal population and (ii) a deduplicated archive of high-scoring policies encountered during search, both scored on $\mathcal{D}_{\mathrm{val}}$ under a common protocol. The deployed policy is then given by

$$\theta^\star \in \arg\max_{\theta \in \mathcal{A}} \widehat{J}(\theta; \mathcal{D}_{\mathrm{val}}). \tag{12}$$

Summary. In summary, ActivationSMC carries a single population of complete policies through M stages of increasing fidelity. Each stage defines a reward-tilted target π_m, connected to the previous stage by an adaptive geometric bridge whose step $s \to s'$ is chosen via ESS control to avoid weight collapse. Diversity is restored by systematic resampling and MH mutation with a shared proposal family q, where only the acceptance target changes with s. Deployment removes multi-fidelity bias by re-evaluating a shortlist from the terminal population and an archive of visited policies on $\mathcal{D}_{\mathrm{val}}$. Algorithm 1 summarizes the full procedure of ActivationSMC.

4 Numerical Results

Experimental Setup. We evaluate ActivationSMC using scalar task rewards, including non-differentiable metrics, without requiring gradients through generation or scoring. Our experiments cover three settings: text-to-text alignment on TruthfulQA (Lin et al., 2022) (Section 4.1), image-to-text alignment on R-Bench (Wu et al., 2024a) and AMBER (Wang et al., 2024) (Section 4.2), and text-to-image alignment with SD3.5 Large Turbo (Section 4.3). We study CAA, COLD-Steer,

[1] See Appendix C.3 for proposal details.

Operator	Tuning	Multiple-Choice				Open-Ended Generation		
		MC1	MC2	MC3	MC Avg.	TRUE	INFO	TRUE×INFO
LLaMA-3.1-8B-Instruct								
Unsteered	–	34.15	53.32	27.02	38.16	82.32	58.60	48.24
Spherical Steering	Native	49.95	68.51	41.05	53.17	82.05	66.57	54.63
	ActivationSMC	**62.89**	**77.34**	**54.94**	**65.06**	84.50	77.03	65.09
COLD-Steer	Native	31.45	55.78	32.01	39.75	75.31	68.15	51.32
	ActivationSMC	**35.94**	**59.24**	**33.41**	**42.86**	77.79	67.35	52.40
AUSteer	Native	43.86	59.97	31.58	45.14	83.64	76.34	63.85
	ActivationSMC	**44.06**	**62.84**	**33.96**	**46.95**	88.89	81.29	72.25
Qwen2.5-7B-Instruct								
Unsteered	–	35.87	54.95	26.62	39.15	83.65	88.93	74.40
Spherical Steering	Native	48.71	66.90	39.16	51.59	88.02	88.44	77.84
	ActivationSMC	**53.12**	**71.03**	**41.84**	**55.33**	83.13	93.03	77.33
COLD-Steer	Native	39.85	58.72	32.01	43.53	87.64	88.92	77.93
	ActivationSMC	**46.08**	**62.87**	**32.34**	**47.10**	85.77	89.35	76.64
AUSteer	Native	35.67	55.34	27.06	39.36	83.11	87.95	73.10
	ActivationSMC	**37.50**	**59.92**	**29.27**	**42.23**	78.11	95.59	74.46

Table 2: TruthfulQA results (%) comparing native tuning and ActivationSMC for the steering operators in Table 1. Bold marks the higher value within each operator–model pair. Native Spherical Steering results are taken from You et al. (2026); the other native baselines are evaluated using the same greedy tuning procedure. Comparable CAA results for these models are unavailable in the cited literature.

AUSteer, and Spherical Steering (Table 1), comparing ActivationSMC with their native tuning procedures and including unsteered models as references. Native tuning runs each operator's published procedure, often greedy or decoupled across layers, restricted to the same number of intervention layers and the same data as ActivationSMC. Each dataset is split into a prototype set for constructing steering signals, a search set for tuning, and a disjoint test set on which all results are reported. Model weights remain frozen throughout. We additionally compare ActivationSMC with Adam (Kingma & Ba, 2014) in Appendix B.1 and with budget-matched simulated annealing and Bayesian optimization in Appendix B.2. We report task performance and generation quality and diversity, and further examine the effects of intervention-layer count, validation schedule, and particle count (Appendices B.3 and B.4). Although native tuning is sometimes cheaper at $K{=}1$, its cost grows with the number of intervention layers while ActivationSMC's remains fixed, making ActivationSMC increasingly efficient as K increases. Detailed NFE comparisons are provided in Appendix A.4. Full experimental settings are provided in Appendix A.

4.1 Text-to-Text Alignment – TruthfulQA

We evaluate COLD-Steer, AUSteer, and Spherical Steering on TruthfulQA with LLaMA-3.1-8B-Instruct and Qwen2.5-7B-Instruct, comparing native tuning against ActivationSMC. For each steering operator, ActivationSMC jointly optimizes the selected layer tuple $\mathcal{L}$ and its local blocks $\theta_{1:K}$ while keeping language-model weights and steering signals fixed. We compare all steered configurations to unsteered models and, for Spherical Steering, to the reference results of You et al. (2026). Prototype construction and policy search use only non-test data, with held-out examples reserved exclusively for evaluation.

Table 2 reports MC1 (best truthful outranks all false), MC2 (normalized mass on truthful), MC3 (fraction truthful outranking best false) and MC Avg., plus open-ended TRUE/INFO judge probabilities and TRUE×INFO (per-fold product, not joint fraction). Our reward emphasizes multiple-choice performance while also incorporating open-ended generation quality. ActivationSMC generally improves its native counterpart: on LLaMA, Spherical 53.17→65.06 MC Avg. (+11.9pp) and 54.63→65.09 TRUE×INFO (+10.5pp), AUSteer 63.85→72.25 TRUE×INFO while raising MC2; on Qwen, MC Avg. improves for all three families and INFO reaches 95.59%. The gains reflect joint population search preserving competing layer-parameter modes, rather than greedy decoupled selection. Open-ended metrics need not all improve, as optimizing the combined reward allows trade-offs among its components.

4.2 Image-to-Text Alignment

In this task, we evaluate image-conditioned prediction on R-Bench (Wu et al., 2024a) and AMBER (Wang et al., 2024) using LLaVA-1.5-7B and Qwen2.5-VL-7B.

Operator	Tuning	R-Bench		AMBER	
		LLaVA-1.5-7B	Qwen2.5-VL-7B	LLaVA-1.5-7B	Qwen2.5-VL-7B
Unsteered	–	72.6/79.1	68.4/69.9	71.6/41.3	82.3/46.4
Spherical Steering	Native	73.7/80.2	**84.2**/87.7	74.1/39.0	82.4/46.3
	ActivationSMC	**76.4**/82.7	83.9/87.3	**75.7**/36.2	**87.1**/54.2
CAA	Native	76.1/82.8	82.2/86.0	73.4/40.8	81.9/46.0
	ActivationSMC	**76.3**/82.8	**84.3**/88.0	**75.4**/32.7	**88.5**/55.2
COLD-Steer	Native	68.9/80.2	83.6/87.5	66.6/29.8	**87.3**/56.2
	ActivationSMC	**76.2**/82.5	**83.8**/87.7	**76.6**/38.4	87.0/53.5
AUSteer	Native	74.6/80.5	78.7/81.9	74.6/37.7	80.7/44.8
	ActivationSMC	**76.2**/82.0	**78.7**/82.2	**79.5**/44.7	**85.6**/50.0

Table 3: Single-layer ($K = 1$) held-out accuracy/F1 (%) on R-Bench and AMBER. Bold marks the higher accuracy between native tuning and ActivationSMC for each operator, benchmark, and model. Full confidence intervals are provided in Table 10 to reduce clutter. In the two settings where native tuning achieves higher accuracy, the differences are within the 95% confidence intervals. In contrast, many of ActivationSMC's improvements remain substantial relative to their confidence intervals, providing stronger evidence of meaningful gains.

Table 3 reports test data accuracy and F1 score for the unsteered models, the native operator-specific tuning procedures, and the configurations selected by ActivationSMC. We first consider the standard single-layer setting ($K = 1$), where each procedure selects one intervention location and its associated steering parameters. ActivationSMC improves held-out accuracy in 14 of the 16 dataset–model–operator combinations, with a mean improvement of 3.01 percentage points and a median improvement of 2.05 points.

The advantage of ActivationSMC widens sharply when the operators are applied at multiple layers. At $K = 4$, the mean improvement over native tuning increases from 3.01 to 9.73 percentage points, while the median improvement more than doubles from 2.05 to 5.51 points. This widening gap supports our central motivation: as additional intervention sites introduce coupled layer and parameter choices, joint population-based search becomes increasingly valuable. Complete $K = 4$ results, including 95% confidence intervals, are provided in Appendix B.3.

Figure 2 shows representative failure modes that drive the gains in Table 3. Across four panels spanning R-Bench / AMBER, Qwen2.5-VL-7B / LLaVA-1.5-7B, and COLD-Steer, Spherical Steering, CAA and AUSteer, the native single-layer tuning (red, *published tuning*) hallucinates—e.g., camera not in front of curtain, four people, a child sitting—while the ActivationSMC policy (green) for the *same* operator and evaluation budget recovers the ground-truth answer.

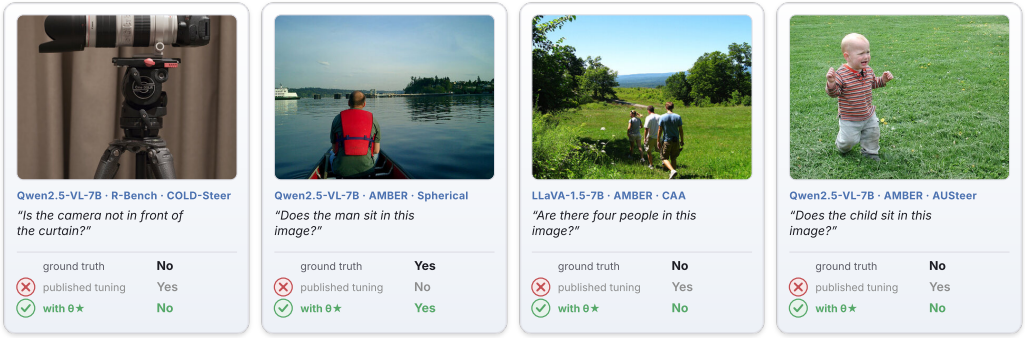


Figure 2: Native $K{=}1$ tuning (red, incorrect) vs. ActivationSMC (green, correct) on R-Bench/AMBER. Same operator and budget; SMC rescues collapsed cases across COLD-Steer, Spherical, CAA and AUSteer.

We additionally ablate three components of the search procedure: the number of particles, the validation schedule, and the number of intervention layers in Appendix B.5. Larger particle populations generally improve performance, consistent with the benefit of broader coverage of the policy space. The number of intervention layers produces the clearest distinction between the search procedures: native tuning becomes substantially less reliable as additional intervention sites are intro-

duced, whereas ActivationSMC retains much more performance and is faster. Full ablation results are provided in Appendix B.

4.3 TEXT-TO-IMAGE ALIGNMENT

We evaluate text-to-image alignment, quality, and diversity on SD3.5 Large Turbo, whose architecture follows Esser et al. (2024). For CAA, COLD-Steer, AUSteer, and Spherical Steering, ActivationSMC jointly optimizes two intervention layers and their local parameters while keeping model weights frozen. Policies are selected on the search split and evaluated on held-out prompts against unsteered and native baselines using identical prompts, seeds, and generation settings.

Soft-TIFA AM/GM (Kamath et al., 2025) measure prompt adherence through arithmetic/geometric means of per-requirement VQA probabilities, followed by averaging across images. GM penalizes poorly satisfied requirements more strongly. **HPSv3** (Ma et al., 2025) estimates human preference for prompt–image pairs. **LPIPS** (Zhang et al., 2018) measures within-prompt diversity through average pairwise perceptual distances across random seeds. Higher Soft-TIFA and HPSv3 indicate better alignment and preference, whereas higher LPIPS indicates greater variation and should be interpreted alongside generation quality.

Table 4 shows improvements over native tuning in all three alignment and preference scores across the four operators. LPIPS decreases for Spherical Steering and CAA but increases slightly for COLD-Steer and AUSteer. These results are consistent with a possible trade-off: favoring higher-scoring outputs may reduce variation across random seeds for the same prompt, so alignment and preference gains need not be accompanied by higher diversity.

		Text–Image Alignment		Human Preference	Diversity
Operator	Tuning	Soft-TIFA AM↑	Soft-TIFA GM↑	HPSv3↑	LPIPS
Unsteered	–	67.45	20.34	6.2213	0.7259
Spherical Steering	Native	77.66	41.09	7.4347	**0.7112**
	ActivationSMC	**84.52**	**61.05**	**7.4820**	0.6941
CAA	Native	69.89	28.73	6.5673	**0.7159**
	ActivationSMC	**79.52**	**58.34**	**7.4123**	0.7056
COLD-Steer	Native	72.63	35.85	6.6331	0.7201
	ActivationSMC	**82.82**	**57.93**	**7.6902**	**0.7221**
AUSteer	Native	73.42	32.10	6.5067	0.7089
	ActivationSMC	**83.90**	**59.61**	**7.6324**	**0.7096**

Table 4: Text-to-image generation results on SD3.5 Large Turbo. We compare native tuning and ActivationSMC for Spherical Steering, CAA, COLD-Steer, and AUSteer. Soft-TIFA AM and GM (Kamath et al., 2025), reported as percentages, measure average and compositional prompt adherence. HPSv3 (Ma et al., 2025) estimates human preference. Bold marks the higher alignment or preference score within each operator pair. LPIPS (Zhang et al., 2018) measures within-prompt perceptual diversity across random seeds; larger values indicate greater variation, rather than necessarily better generation quality.

5 CONCLUSION

We introduced ActivationSMC, a derivative-free black-box framework that optimizes activation steering configurations as joint inference over where and how strongly to intervene. It maintains a population of full multi-layer configurations evolved via reweighting, resampling, and mutation toward reward-tilted Gibbs targets, with a multi-fidelity schedule and ESS-controlled tempering that uses cheap noisy estimates for broad early exploration and larger subsets for precise late refinement. Without updating model weights, it configures existing prominent steering operators such as CAA, COLD-Steer, AUSteer, and Spherical Steering, rescuing collapsed baselines by up to 28.9 percentage points and improving compositional text-to-image alignment, with gains widening as coupling grows at higher K across TruthfulQA, R-Bench, AMBER, and SD3.5. Potential future work includes uncertainty-aware adaptive fidelity, input-dependent and temporally varying steering policies, and auditing the transfer and safety of selected configurations across tasks, models, and deployment conditions.

AI USE STATEMENT

In this work, we used AI tools for no tasks with required disclosure. We have not used AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in writing proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, translation, cleaning and reformatting datasets, qualitative and thematic data analysis, or interpreting results, and these tasks are not applicable beyond what is reported in the paper.

Additionally, we used AI tools for tasks with recommended disclosure: drafting parts of the research paper, editing the paper to improve readability, summarizing and analyzing existing literature, identifying relevant literature, and sourcing / searching for information. Specifically, AI was used to help polish writing, to help improve code efficiency/readability, and to assist with literature search and summarization. We have reviewed AI-assisted work. AI-generated drafts were manually rewritten and verified, literature summaries and suggested references were manually checked against original sources for accuracy, and all final text, claims, and artifacts were validated by the authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Activation steering can improve model behavior without updating model weights, but the same techniques can also be used to induce undesirable behavior, weaken existing safeguards, or manipulate model outputs in ways that are difficult for users to detect. ActivationSMC may increase this capability by making steering policies easier to optimize across several layers and task-specific rewards. Our experiments are focused on alignment and benchmark performance, but success on these evaluations does not establish general safety or reliability outside the tested settings. The selected policies should therefore be evaluated for unintended behavioral changes, capability degradation, and misuse before deployment. We encourage responsible release and use of steering configurations, particularly when interventions affect safety-sensitive or user-facing systems.

REPRODUCIBILITY STATEMENT

We provide the mathematical formulation of ActivationSMC, definitions of each steering operator, and details of the weighting, resampling, mutation, archiving, and final-selection procedures. The appendices report the search spaces, evaluation budgets, particle counts, ESS targets, resampling schedules, mutation settings, random seeds, and full-fidelity selection protocols used in each experiment. We also report uncertainty across independent runs and index convergence by completed policy evaluations to make comparisons across population sizes meaningful. Additionally, all code is submitted as supplementary materials.

REFERENCES

Tetiana Bas and Krystian Novak. What can we actually steer? a multi-behavior study of activation control, 2026. URL https://arxiv.org/abs/2511.18284.

Alexandros Beskos, Ajay Jasra, Nikolas Kantas, and Alexandre H. Thiéry. On the convergence of adaptive sequential Monte Carlo methods. *The Annals of Applied Probability*, 26(2), 2016. doi: 10.1214/15-AAP1113. URL https://arxiv.org/abs/1306.6462.

Nicolas Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

Pierre Del Moral. *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.

Pierre Del Moral. *Mean Field Simulation for Monte Carlo Integration*. Chapman and Hall/CRC, 2013.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006. doi: 10.1111/j.1467-9868.2006.00553.x.

Pierre Del Moral, Frédéric Patras, and Sylvain Rubenthaler. Tree based functional expansions for Feynman–Kac particle models. *The Annals of Applied Probability*, 19(2):778–825, 2009.

Randal Douc and Eric Moulines. Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *The Annals of Statistics*, 36(5):2344–2376, 2008.

Randal Douc, Olivier Cappé, and Eric Moulines. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 64–69, 2005.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024. URL https://arxiv.org/abs/2403.03206.

Zijian Feng, Tianjiao Li, Zixiao Zhu, Hanzhang Zhou, Junlang Qian, Li Zhang, Jia Jim Deryl Chua, Lee Onn Mak, Gee Wah Ng, and Kezhi Mao. Fine-grained activation steering: Steering less, achieving more. *arXiv preprint arXiv:2602.04428*, 2026. doi: 10.48550/arXiv.2602.04428. URL https://arxiv.org/abs/2602.04428. ICLR 2026.

Alex Ferrando, Xavier Suau, Jordi Gonzàlez, and Pau Rodriguez. Dynamically scaled activation steering. *arXiv preprint arXiv:2512.03661*, 2025. doi: 10.48550/arXiv.2512.03661. URL https://arxiv.org/abs/2512.03661.

Soham Gadgil, Chris Lin, and Su-In Lee. Where to steer: Input-dependent layer selection for steering improves LLM alignment. *arXiv preprint arXiv:2604.03867*, 2026. doi: 10.48550/arXiv.2604.03867. URL https://arxiv.org/abs/2604.03867.

Mathieu Gerber, Nicolas Chopin, and Nick Whiteley. Negative association, ordering and convergence of resampling methods. *The Annals of Statistics*, 47(4):2236–2260, 2019.

Michael B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

Gabriel Grand, Joshua B. Tenenbaum, Vikash K. Mansinghka, Alexander K. Lew, and Jacob Andreas. Self-steering language models. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=XvCBtm5PgF.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97.

Brandon Hsu, Daniel Beaglehole, Adityanarayanan Radhakrishnan, and Mikhail Belkin. Contextual linear activation steering of language models. *arXiv preprint arXiv:2604.24693*, 2026. doi: 10.48550/arXiv.2604.24693. URL https://arxiv.org/abs/2604.24693.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2303.11897.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. GenEval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025. URL https://arxiv.org/abs/2512.16853.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Wen Lai, Alexander Fraser, and Ivan Titov. Joint localization and activation editing for low-resource fine-tuning. *arXiv preprint arXiv:2502.01179*, 2025. doi: 10.48550/arXiv.2502.01179. URL https://arxiv.org/abs/2502.01179. ICML 2025.

A. Lecchini-Visintini, J. Lygeros, and J. Maciejowski. Simulated annealing: Rigorous finite-time guarantees for optimization on continuous domains, 2007. URL https://arxiv.org/abs/0709.2989.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Oi47wc10sm.

Alexander K. Lew, Zhi-Xuan Tan, Gabriel Grand, and Vikash K. Mansinghka. Sequential monte carlo steering of large language models using probabilistic programs. *arXiv preprint arXiv:2306.03081*, 2023. doi: 10.48550/arXiv.2306.03081. URL https://arxiv.org/abs/2306.03081.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander K. Lew, Tim Vieira, and Timothy J. O'Donnell. Syntactic and semantic control of large language models via sequential monte carlo. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xoXn62FzD0.

Yuhang Ma, Yunhao Shui, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2508.03789.

Christian A Naesseth, Fredrik Lindsten, and Thomas B Schön. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):187–306, 2019.

Muhammad Faishal Adly Nelwan and Alfan Farizki Wicaksono. Deployable per-instance multi-layer activation steering for large language models, 2026. URL https://arxiv.org/abs/2608.08829.

Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Arnaud Dapogny, Alasdair Newson, and Matthieu Cord. Learning to steer: Input-dependent steering for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5341. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ea491e2d1c46686b8db5cd11154f5d2c-Abstract-Conference.html.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Optimal model management for multifidelity Monte Carlo estimation. *SIAM Journal on Scientific Computing*, 38(5):A3163–A3194, 2016.

Isha Puri, Shivchander Sudalairaj, Guangxuan Xu, Abhishek Bhandwaldar, Kai Xu, and Akash Srivastava. Rollout roulette: A probabilistic inference approach to inference-time scaling of LLMs using particle-based monte carlo methods. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5227. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e55c675d3230dbc3bf24c986d6685632-Abstract-Conference.html.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

John T. Robertson, Jianing Zhu, Haris Vikalo, and Zhangyang Wang. When is rank-1 steering cheap? geometry, granularity, and budgeted search. *arXiv preprint arXiv:2605.16362*, 2026. doi: 10.48550/arXiv.2605.16362. URL https://arxiv.org/abs/2605.16362.

Kartik Sharma and Rakshit S. Trivedi. COLD-Steer: Steering large language models via in-context one-step learning dynamics. *arXiv preprint arXiv:2603.06495*, 2026. doi: 10.48550/arXiv.2603.06495. URL https://arxiv.org/abs/2603.06495. ICLR 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Hieu M. Vu and Tan M. Nguyen. Angular steering: Behavior control via rotation in activation space. *arXiv preprint arXiv:2510.26243*, 2025. doi: 10.48550/arXiv.2510.26243. URL https://arxiv.org/abs/2510.26243. NeurIPS 2025 Spotlight.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation, 2024. URL https://arxiv.org/abs/2311.07397.

Mingrui Wu, Jiayi Ji, Oucheng Huang, Jiale Li, Yuhang Wu, Xiaoshuai Sun, and Rongrong Ji. Evaluating and analyzing relationship hallucinations in large vision-language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53553–53570, 2024a. URL https://proceedings.mlr.press/v235/wu24l.html.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024b.

Zejia You, Chunyuan Deng, and Hanjie Chen. Spherical steering: Geometry-aware activation rotation for language models. *arXiv preprint arXiv:2602.08169*, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://arxiv.org/abs/1801.03924.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024. URL https://arxiv.org/abs/2404.17546.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Experimental Details

This appendix lists the search settings used in Section 4. Unless stated otherwise, all settings are shared across operators within a setting and fixed before search.

A.1 Search Configuration

Table 5 reports the number of intervention layers K, the population size N, the evaluation schedule, and the resulting number of policy evaluations per run. Each search proceeds through $M = 4$ stages of nested evaluation subsets $\mathcal{B}_1 \subseteq \mathcal{B}_2 \subseteq \mathcal{B}_3 \subseteq \mathcal{B}_4$, with a fixed number of rounds per stage. Every round scores each particle once on the current subset, so a run uses $N \sum_m R_m$ policy evaluations in total.

	Text-to-text (TruthfulQA)	Image-to-text (AMBER/R-Bench)	Text-to-image (SD3.5)
Intervention layers K	2	1	2
Particles N	64	96	32
Subset sizes $\lvert\mathcal{B}_{1:4}\rvert$	16, 32, 64, all	24, 48, 96, 384	16, 32, 64, all
Rounds per stage $R_{1:4}$	2, 2, 2, 2	2, 2, 2, 1	2, 2, 2, 2

Table 5: Search configuration per setting. "all" denotes the full search split. The particle-count and layer-count ablations (Appendix B.4) vary $N \in \{12, 24, 48, 96\}$ and $K \in \{1, \ldots, 4\}$ one at a time from the image-to-text defaults.

Effective sample size. At each round the reweighting temperature is chosen by bisection so that the effective sample size equals a target fraction $\rho = 0.6$ of the population, i.e., $\mathrm{ESS} = 0.6N$. These values are used in all experiments.

A.2 Mutation

Mutation uses the global and local proposals defined in Appendix C.3, with the rates in Table 6. These rates are shared across all settings and operators.

Symbol	Meaning	Value
ρ	Target ESS fraction	0.60
p_g	Global-proposal probability	0.25
p_{layer}	Layer / integer perturbation probability	0.60
σ	Base step fraction	0.15
γ	Step-size decay per round	0.85
p_{cat}	Categorical switch probability	0.30

Table 6: Hyperparameters shared by all ActivationSMC runs.

A.3 Native Tuning

Native tuning runs each operator's published procedure on the same prototype and search data as ActivationSMC, restricted to the same number of intervention layers K. These procedures are frequently greedy or decouple layer choice from parameter choice: COLD-Steer evaluates a fixed grid over layer and strength (Sharma & Trivedi, 2026); AUSteer sweeps its unit count and strength jointly as a grid search (Feng et al., 2026); Spherical Steering selects a layer with an ungated sweep and then tunes its strength and gate at that layer (You et al., 2026); and CAA selects a layer and then its multiplier (Rimsky et al., 2024).

A.4 Neural Function Evaluations

We compare the neural function evaluations (NFEs) required by different methods for the image-to-text experiments (see Figure 3). We focus on this multimodal setting because it provides a particularly informative comparison. NFEs offer a more hardware-independent measure of computational

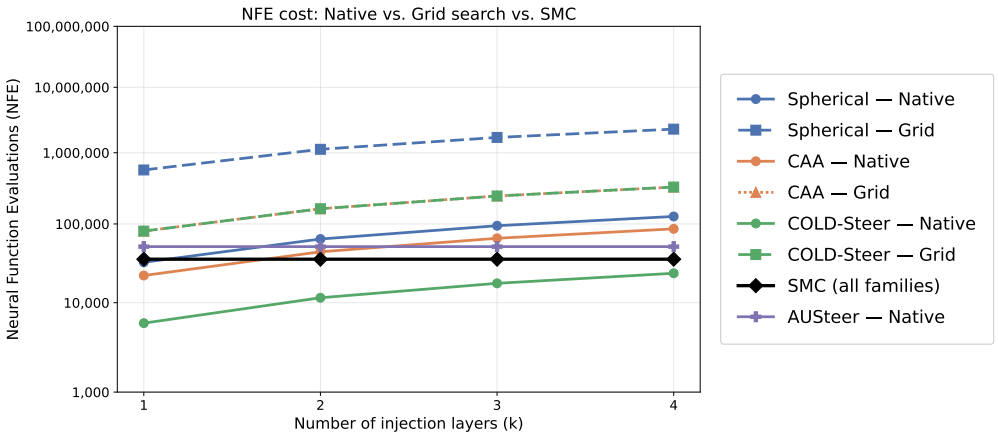


Figure 3: NFE comparison across methods as a function of layers intervened on.

cost than wall-clock time, which can vary with hardware configurations and parallelization across GPUs. AUSteer uses a fixed 50,190 NFEs per run, while ActivationSMC uses a fixed 34,560. Notably, as the number of parameters increases, the NFE gap between native approaches and ActivationSMC narrows. In some settings, ActivationSMC achieves better performance while requiring fewer NFEs.

B Complementary Experiments

B.1 ActivationSMC versus Adam

For the experiment in Section 4.1, we compare ActivationSMC with a gradient-based baseline that optimizes continuous steering parameters using Adam (Kingma & Ba, 2015) on the same search split. Table 7 reports held-out performance for the unsteered model, Adam-optimized steering, and ActivationSMC, including the individual multiple-choice metrics, their average (MC Avg.), and open-ended generation metrics.

Model	Method	Multiple-Choice				Open-Ended Generation		
		MC1	MC2	MC3	MC Avg.	TRUE	INFO	TRUE×INFO
LLaMA-3.1-8B-Instruct	Baseline	34.15	53.32	27.02	38.16	82.32	58.60	48.24
LLaMA-3.1-8B-Instruct	Trainable Steering	40.59	61.20	34.13	45.31	93.03	50.61	47.08
LLaMA-3.1-8B-Instruct	**ActivationSMC (Ours)**	**62.89**	**77.34**	**54.94**	**65.06**	84.50	77.03	65.09

Table 7: Comparison of Adam-optimized steering and ActivationSMC search.

The results show different performance–preservation trade-offs for Adam and ActivationSMC. Population search and occasional global layer moves provide mechanisms for exploring configurations beyond a single local optimization trajectory, but these results alone do not establish that Adam reached a local optimum or that ActivationSMC found a global one. A further practical distinction is reward access: TRUE×INFO requires generated answers and external judges, and ordinary backpropagation does not differentiate through discrete decoding. Gradient-based optimization of such a reward is possible with suitable estimators or relaxations, whereas ActivationSMC can use its scalar value directly. Both approaches still incur the cost of the required rollouts and evaluations.

B.2 ActivationSMC vs. Simulated Annealing and Bayesian Optimization

To isolate how much of ActivationSMC's advantage in Section 4.2 comes from the SMC search itself, rather than simply from spending a fixed evaluation budget searching *some* hyperparameter configurations, we extend the experiment in Section 4.2 with two additional budget-matched baselines: a single-trajectory simulated annealing (SA) chain and a Gaussian-process Bayesian optimization (BO) baseline. Both are matched to ActivationSMC's exact evaluation budget, but explore the search space differently from ActivationSMC and from each other: SA proposes and accepts/rejects

Family	Native	ActivationSMC	SA	BO
Spherical	74.0	**76.7**	73.6	75.9
CAA	77.1	**77.5**	73.7	76.5
COLD-Steer	69.9	**76.4**	73.2	73.4

Table 8: Accuracy (%) (R-Bench, LLaVA-1.5-7B): default tuning (Native) vs. SMC search vs. single-trajectory simulated annealing (SA) vs. Bayesian optimization (BO), matched to the same evaluation budget.

Dataset	Model	Operator	Native tuning (%)	ActivationSMC (%)	Δ (pp)
AMBER	LLaVA-1.5-7B	CAA	61.86 [61.06, 62.66]	**74.44** [73.72, 75.15]	+12.58
AMBER	LLaVA-1.5-7B	COLD-Steer	54.30 [53.48, 55.12]	**73.85** [73.12, 74.56]	+19.55
AMBER	LLaVA-1.5-7B	Spherical Steering	69.02 [68.26, 69.78]	**76.12** [75.41, 76.81]	+7.10
AMBER	Qwen2.5-VL-7B	CAA	85.13 [84.54, 85.70]	**88.24** [87.70, 88.76]	+3.11
AMBER	Qwen2.5-VL-7B	COLD-Steer	59.32 [58.51, 60.13]	**88.22** [87.68, 88.74]	+28.90
AMBER	Qwen2.5-VL-7B	Spherical Steering	83.19 [82.56, 83.79]	**86.52** [85.95, 87.07]	+3.33
R-Bench	LLaVA-1.5-7B	CAA	68.90 [67.86, 69.92]	**72.81** [71.81, 73.79]	+3.92
R-Bench	LLaVA-1.5-7B	COLD-Steer	64.93 [63.86, 65.98]	**74.43** [73.45, 75.39]	+9.50
R-Bench	LLaVA-1.5-7B	Spherical Steering	72.78 [71.78, 73.75]	**75.09** [74.11, 76.03]	+2.31
R-Bench	Qwen2.5-VL-7B	CAA	82.42 [81.21, 83.56]	**83.87** [82.71, 84.97]	+1.46
R-Bench	Qwen2.5-VL-7B	COLD-Steer	58.73 [57.63, 59.82]	**83.92** [83.09, 84.72]	+25.20
R-Bench	Qwen2.5-VL-7B	Spherical Steering	**83.09** [81.86, 84.26]	82.85 [81.61, 84.03]	−0.24

Table 9: Four-layer ($K = 4$) held-out accuracy on R-Bench and AMBER. Brackets contain 95% confidence intervals, and bold denotes the higher point estimate in each row. Native tuning extends each operator's original tuning procedure to four intervention layers, whereas ActivationSMC jointly searches over the four intervention locations and their associated parameters. The final column reports the ActivationSMC improvement in percentage points. AUSteer not shown since it is already multi-layer.

along a single MH chain, carrying no notion of an effective sample size or population-level diversity, while BO fits a surrogate model over every evaluation seen so far and selects each round's batch by expected improvement over a large, freshly-sampled candidate pool. We run all three methods with 96 particles on LLaVA-1.5-7B, fold 0, on R-Bench, across the Spherical, CAA, and COLD-Steer families.

ActivationSMC outperforms both budget-matched baselines on every family we test. This suggests that SMC's advantage over the default tuning is not merely a byproduct of spending compute on *some* search, but comes from maintaining a weighted population and adaptively resampling.

B.3 Multi-Layer Image-to-Text Results

We repeat the image-to-text evaluation with each steering operator applied jointly at $K = 4$ intervention layers. This experiment is designed as a stress test of multi-layer optimization rather than as a claim that increasing the number of intervention layers necessarily improves absolute performance. In contrast to the $K = 1$ setting, the tuning procedure must jointly select four intervention locations and their associated operator hyperparameters.

Table 9 reports the complete results. ActivationSMC improves held-out accuracy in 11 of the 12 dataset–model–operator combinations. In the remaining setting, its accuracy differs from native tuning within the error bounds of the model (similar to $K = 1$). The mean improvement is 9.73 percentage points, and the median improvement is 5.51 points. Per dataset, the mean improvement is 12.43 points on AMBER and 7.03 points on R-Bench.

The larger relative gains at $K = 4$ arise primarily because ActivationSMC is more robust to the expanded search space. These results suggest that the benefit of joint population-based optimization grows as additional intervention layers introduce more strongly coupled layer and parameter choices.

Dataset	Model	Operator	Native tuning (%)	ActivationSMC (%)	Δ (pp)
AMBER	LLaVA-1.5-7B	AUSteer	74.56 [73.83, 75.30]	**79.51** [78.84, 80.16]	+4.95
AMBER	LLaVA-1.5-7B	CAA	73.44 [72.70, 74.17]	**75.37** [74.66, 76.07]	+1.93
AMBER	LLaVA-1.5-7B	COLD-Steer	66.62 [65.85, 67.39]	**76.58** [75.87, 77.27]	+9.95
AMBER	LLaVA-1.5-7B	Spherical Steering	74.06 [73.32, 74.77]	**75.71** [75.00, 76.43]	+1.65
AMBER	Qwen2.5-VL-7B	AUSteer	80.68 [80.04, 81.32]	**85.62** [85.05, 86.20]	+4.94
AMBER	Qwen2.5-VL-7B	CAA	81.88 [81.24, 82.51]	**88.48** [87.96, 89.00]	+6.60
AMBER	Qwen2.5-VL-7B	COLD-Steer	**87.31** [86.77, 87.86]	86.97 [86.42, 87.53]	−0.34
AMBER	Qwen2.5-VL-7B	Spherical Steering	82.40 [81.77, 83.03]	**87.06** [86.50, 87.61]	+4.66
R-Bench	LLaVA-1.5-7B	AUSteer	74.57 [73.62, 75.54]	**76.22** [75.27, 77.17]	+1.65
R-Bench	LLaVA-1.5-7B	CAA	76.15 [75.20, 77.12]	**76.33** [75.37, 77.28]	+0.18
R-Bench	LLaVA-1.5-7B	COLD-Steer	68.88 [67.86, 69.91]	**76.19** [75.24, 77.14]	+7.31
R-Bench	LLaVA-1.5-7B	Spherical Steering	73.75 [72.78, 74.74]	**76.38** [75.45, 77.31]	+2.63
R-Bench	Qwen2.5-VL-7B	AUSteer	78.71 [77.80, 79.61]	**78.72** [77.82, 79.63]	+0.01
R-Bench	Qwen2.5-VL-7B	CAA	82.21 [81.38, 83.06]	**84.31** [83.52, 85.10]	+2.09
R-Bench	Qwen2.5-VL-7B	COLD-Steer	83.63 [82.80, 84.42]	**83.79** [83.01, 84.60]	+0.16
R-Bench	Qwen2.5-VL-7B	Spherical Steering	**84.18** [83.38, 84.97]	83.91 [82.89, 84.50]	−0.27

Table 10: Single-layer ($K = 1$) held-out accuracy on R-Bench and AMBER, ActivationSMC with 96 particles. Brackets contain 95% confidence intervals, and bold denotes the higher point estimate in each row. Native tuning is each operator's original, fixed single-layer configuration (or multi-layer in AUSteer), whereas ActivationSMC searches over the intervention layer and its associated parameters. The final column reports the ActivationSMC improvement in percentage points.

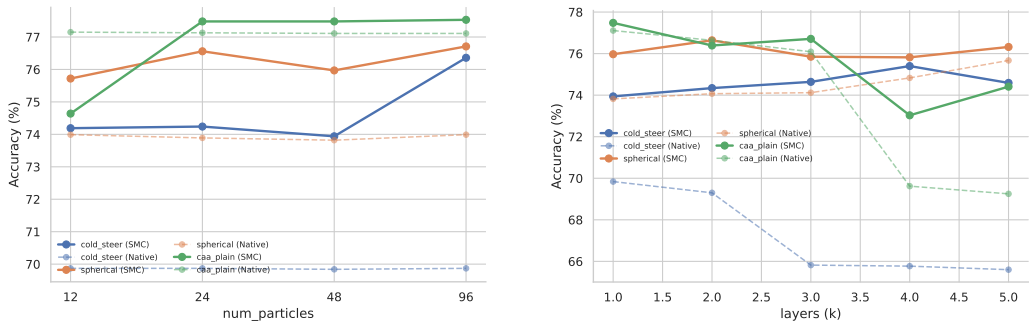
B.4 Population and Layer-Count Ablations on R-Bench

We study how ActivationSMC scales with the number of particles N and intervention layers K for COLD-Steer, CAA, and Spherical Steering on R-Bench. We vary one quantity at a time while holding the remaining search configuration fixed. AUSteer is omitted from these plots because its configuration does not depend on K as a frozen parameter. We measure search progress using completed configuration evaluations rather than SMC iterations, since increasing the population size increases the cost of each iteration.

Effect of population size. Figure 4a shows performance as the number of particles increases. Larger populations generally improve performance across the three steering operators, although the magnitude of the improvement varies by operator. Increasing N allows the search to maintain more competing layer–parameter configurations and reduces the likelihood that promising regions are lost during resampling and mutation. This improvement represents a direct compute–performance tradeoff: larger populations provide broader coverage of the search space but require more configuration evaluations per iteration.

Effect of intervention-layer count. Figure 4b compares ActivationSMC with each operator's established tuning procedure as the number of intervention layers increases. Additional layers expand the discrete space of layer combinations, introduce independently parameterized interventions, and create dependencies between choices at different depths. Consequently, configurations selected by tuning layers or parameters separately become less reliable as K grows. Across the three operators, ActivationSMC is generally more robust to this increase in search-space complexity, and its advantage over the established procedures tends to widen with the number of intervention layers. The size of this effect remains operator-dependent.

These results support the motivation for joint population-based search. Each ActivationSMC particle represents a complete multi-layer configuration, allowing selection to act on compatible combinations of intervention locations and local parameters. The results do not imply that adding layers is always beneficial; rather, they show that ActivationSMC more effectively searches the increasingly coupled configuration space created by multi-layer intervention.



(a) Number of particles N.

(b) Number of intervention layers K.

Figure 4: Scaling behavior of ActivationSMC for COLD-Steer, CAA, and Spherical Steering on R-Bench. **Left:** Performance as the particle population increases. Larger populations provide broader coverage of the configuration space, generally improving performance at additional computational cost. **Right:** Performance as the number of intervention layers increases. ActivationSMC is generally more robust than the established operator-specific tuning procedures as additional coupled layer and parameter choices are introduced.

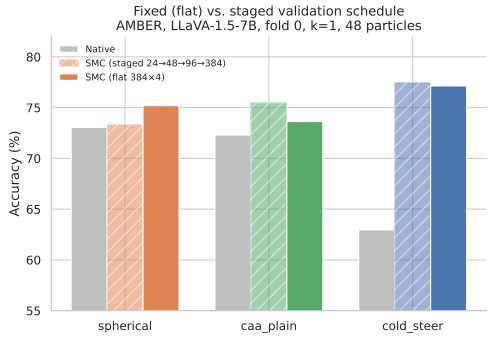


Figure 5: Effect of replacing ActivationSMC's staged validation schedule ($24 \rightarrow 48 \rightarrow 96 \rightarrow 384$ items) with a flat schedule that evaluates every stage on the full 384-item split, for COLD-Steer, CAA, and Spherical Steering on a subset of AMBER (LLaVA-1.5-7B, fold 0). The effect is operator-dependent—Spherical Steering improves, CAA degrades, and COLD-Steer is roughly unchanged—suggesting steering methods differ in their sensitivity to noisy early-stage signal rather than uniformly benefiting from higher-fidelity evaluation throughout search.

B.5 Validation-Schedule Ablation on AMBER

ActivationSMC evaluates each particle on a validation subset whose size grows across search stages ($24 \rightarrow 48 \rightarrow 96 \rightarrow 384$ items), so early generations are scored on a cheap, noisy signal and only the final generation is scored on the full validation split. To isolate the effect of this staged-fidelity schedule, we replace it with a flat schedule that evaluates every stage on the full 384-item split, holding the population size (N=48), mutation schedule, and total number of stages fixed. We run this ablation for COLD-Steer, CAA, and Spherical Steering on a subset of AMBER with LLaVA-1.5-7B.

Figure 5 shows that the effect of removing the staged schedule is mixed and operator-dependent. Spherical Steering improves under the flat schedule, while CAA is worse, and COLD-Steer is essentially unchanged. The operator-dependent direction suggests that steering methods differ in how sensitive their search trajectories are to noisy early signal. Some benefit from resolving that noise immediately, while others perform better under the staged schedule's cheap early exploration and only need high-fidelity evaluation once the search has narrowed. Because the flat schedule spends roughly $3.7\times$ more compute per particle evaluation than the staged one, the staged schedule remains the more compute-efficient default; we do not find evidence that abandoning it is broadly beneficial.

C Notation and Supplementary ActivationSMC Details

We use the notation of Section 3. The steering operator and the number of intervention layers K are fixed within each run. The feasible space Θ includes the prescribed layer and parameter constraints. Densities use counting measure on discrete coordinates and Lebesgue measure on continuous search coordinates. The reference density ν_0 defines a probability distribution on Θ. Table 11 shows the notation.

Symbol	Meaning
$L, K, \mathcal{L}$	Model depth, number of intervention layers, and ordered layer tuple.
$\theta = (\mathcal{L}, \theta_{1:K})$	Complete steering policy; θ_j contains the local parameters at layer l_j.
$N, \theta^{(i)}$	Population size and particle i.
$J, \widehat{J}_m$	Population reward and its estimate on $\mathcal{B}_m$.
$m, M, \mathcal{B}_m, S_m$	Evaluation stage, number of stages, evaluation subset, and generation replicates per input.
ν_0, π_m	Reference distribution and stage target.
τ_m	Prescribed inverse temperature at stage m.
$s, s', \pi_{m,s}$	Current and next transition values, and the intermediate target.
w_i, ρ	Normalized particle weight and target ESS fraction.
$q(\theta' \mid \theta)$	Proposal density or probability used in an MH update.
$\mathcal{A}, \theta^\star$	Eligible validation finalists and the selected policy.

Table 11: Notation for the main method. Here m indexes evaluation stages, and s controls progress between consecutive stage targets.

C.1 Optional Preservation Criterion

Let $\widehat{C}(\theta; \mathcal{B})$ estimate a preservation score for which larger values are better. Given a reference value C_0 and allowed degradation $\varepsilon \geq 0$, an optional search score is

$$\widehat{R}(\theta; \mathcal{B}) = \widehat{J}(\theta; \mathcal{B}) - \lambda \left[\frac{C_0 - \varepsilon - \widehat{C}(\theta; \mathcal{B})}{s_C} \right]_+^2, \tag{13}$$

where $[z]_+ = \max(z, 0)$, $\lambda \geq 0$ controls the penalty strength, and $s_C > 0$ sets its scale. The penalty applies when the estimated preservation score falls below $C_0 - \varepsilon$. Setting $\lambda = 0$ recovers the task reward. The population penalized objective R is obtained by replacing $\widehat{J}$ and $\widehat{C}$ with their population counterparts J and C in (13).

When this penalty is enabled, define $\widehat{R}_m(\theta) = \widehat{R}(\theta; \mathcal{B}_m)$ and replace $\widehat{J}_m$ by $\widehat{R}_m$ consistently in the stage targets, incremental weights, and MH acceptance probabilities. The penalty settings are fixed before search.

A hard preservation requirement can also be applied at final selection:

$$\widehat{C}(\theta; \mathcal{D}_{\text{val}}) \geq C_0 - \varepsilon.$$

In this case, $\mathcal{A}$ in (12) contains only finalists satisfying this requirement. If none qualify, we report that no feasible candidate was found. The search penalty and final constraint can be enabled separately; satisfying the empirical constraint does not by itself guarantee preservation on unseen inputs.

C.2 Incremental Reweighting and Adaptive ESS

For a fixed stage m and current particle population, define the incremental scores

$$g_i = \tau_m \widehat{J}_m(\theta^{(i)}) - \tau_{m-1} \widehat{J}_{m-1}(\theta^{(i)}), \tag{14}$$

where the second term is zero when $m = 1$. Both estimates are evaluated at the same policy. Substituting the stage targets into (9) gives

$$w_i = \frac{\exp((s' - s) g_i)}{\sum_{j=1}^N \exp((s' - s) g_j)}.$$

Thus the weights correct for both the changed evaluation subset and inverse temperature, as required by sequential importance sampling (Del Moral et al., 2006). We normalize the log weights using log-sum-exp.

Variational interpretation. Let $u_i = 1/N$. The weights uniquely solve

$$w = \arg\max_{\substack{v_i \geq 0 \\ \sum_i v_i = 1}} \left\{ (s' - s) \sum_{i=1}^{N} v_i g_i - D_{\mathrm{KL}}(v \| u) \right\}. \tag{15}$$

For any probability vector v, direct substitution yields

$$(s' - s) \sum_i v_i g_i - D_{\mathrm{KL}}(v \| u) = \log \left(\frac{1}{N} \sum_i e^{(s'-s) g_i} \right) - D_{\mathrm{KL}}(v \| w).$$

Nonnegativity of KL proves the claim. Comparing the objective at w and u also gives

$$(s' - s) \left(\sum_i w_i g_i - \frac{1}{N} \sum_i g_i \right) \geq D_{\mathrm{KL}}(w \| u) \geq 0.$$

This inequality concerns the incremental scores g_i. It does not assert monotonic improvement of the task reward after mutation or after changing the evaluation subset. The KL acts on particle-selection probabilities and is distinct from a model-output preservation penalty.

ESS adaptation. For a proposed increment $\delta = s' - s \geq 0$, write $Z(\delta) = \sum_i e^{\delta g_i}$. Then

$$\mathrm{ESS}(\delta) = \frac{Z(\delta)^2}{Z(2\delta)}, \qquad \frac{d}{d\delta} \log \mathrm{ESS}(\delta) = 2 \left(\frac{Z'(\delta)}{Z(\delta)} - \frac{Z'(2\delta)}{Z(2\delta)} \right) \leq 0,$$

because $\log Z$ is convex. Since $\mathrm{ESS}(0) = N$, bisection finds the largest allowed increment when a full advance to $s' = 1$ would violate (10) (Beskos et al., 2016). If all g_i coincide, the weights remain uniform and the full advance is admissible. The endpoint temperatures τ_m remain prescribed; ESS selects the intermediate steps. Every advance is followed by resampling and a reset to equal weights.

C.3 Stagewise SMC and Metropolis–Hastings Updates

Fix the evaluation subsets, generation seeds, steering signals, and reward settings. Each $\widehat{J}_m$ is then a fixed function of θ. With bounded stage rewards and a proper reference ν_0, the targets in (7) are well defined. They share the support of ν_0. Reweighting an ideal distribution $\pi_{m,s}$ by the ratio $\pi_{m,s'}/\pi_{m,s}$ gives $\pi_{m,s'}$. A subsequent Markov update that preserves $\pi_{m,s'}$ therefore leaves this new target unchanged, which is the SMC sampler construction (Del Moral et al., 2006).

Resampling. Systematic resampling draws one offset uniformly from $[0, 1/N)$ and places N equally spaced points in the cumulative weight distribution. Particle i has expected offspring count Nw_i, so resampling preserves the weighted empirical distribution in expectation. The offspring can be dependent. The archive is used for final selection and is external to these population updates.

MH invariance. Let $a(\theta, \theta')$ denote the acceptance probability in (11). For fixed target and proposal settings, the accepted transitions satisfy detailed balance:

$$\begin{aligned} &\pi_{m,s'}(\theta) q(\theta' \mid \theta) a(\theta, \theta') \\ &\quad = \min\{\pi_{m,s'}(\theta) q(\theta' \mid \theta), \pi_{m,s'}(\theta') q(\theta \mid \theta')\} \\ &\quad = \pi_{m,s'}(\theta') q(\theta \mid \theta') a(\theta', \theta). \end{aligned}$$

Together with retaining the current policy on rejection, this proves invariance of the MH kernel (Hastings, 1970). For a coordinate update, q denotes the conditional proposal density or probability for the coordinates being changed.

Proposals on the feasible policy space. A policy combines discrete layer indices, integer and categorical parameters, and bounded continuous parameters, and the reward provides only scalar, possibly non-differentiable evaluations. Proposals are therefore derivative-free and defined coordinatewise by type, combining a global component for moving between distant modes with a local component for refinement. The rates below are listed in Table 6.

With probability p_g, a global move redraws the entire configuration, including intervention layers, from ν_0, allowing particles to jump between separated layer–parameter modes (Appendix E) that local moves cannot connect. Otherwise, a local move perturbs each coordinate. Layers move one grid step (± 1, uniformly) with probability p_{layer}, while remaining attached to their parameters, so that a well-tuned local block can be tried at a neighboring depth. A continuous parameter with range $[\underline{c}, \overline{c}]$ receives Gaussian noise with standard deviation $\sigma\gamma^t(\overline{c}-\underline{c})$, where σ is the base step fraction, γ is the step-size decay rate, and t is the cumulative iteration count across all stages; the decay makes early moves coarse while evaluation subsets are small and later moves fine as they grow. Integer parameters move in a random direction by $\max\{1, \text{round}[\sigma\gamma^t(\overline{c}-\underline{c})]\}$ with the same probability p_{layer}, and categorical parameters switch uniformly to another category with probability p_{cat}. Local proposals are clipped to the feasible range. These settings are used in all experiments.

We select proposal components with fixed probabilities and apply the MH correction using the selected component's density q. Each corrected kernel preserves the current target, as does their mixture (Hastings, 1970). All densities use the same search coordinates, with Jacobians included when required. Proposal scales and MH move counts are specified in the experiment configuration.

Intermediate targets generally require both current-stage and previous-stage reward estimates. Repeated MH moves at an unchanged target require no additional reweighting (Del Moral et al., 2006). Kernel invariance alone does not ensure accurate finite-population sampling; the next subsection relates particle-approximation error to optimization error.

C.4 From Distribution Approximation to Optimization

We relate distribution approximation to optimization through target concentration and total-variation sampling error, following the general perspective of finite-temperature stochastic optimization (Lecchini-Visintini et al., 2007). We give a self-contained bound for terminal selection using $\widehat{J}_M$, allowing dependent particles.

Condition on the search data, generation seeds, fixed steering signals, and prescribed targets. All probabilities below are over the remaining randomness of the SMC algorithm.

Theorem C.1 (Optimization error of terminal selection)**.** *Suppose $J^\star = \sup_{\theta\in\Theta} J(\theta) < \infty$ and*

$$\sup_{\theta\in\Theta} \left| \widehat{J}_M(\theta) - J(\theta) \right| \le \epsilon_J, \qquad \epsilon_J \ge 0. \tag{16}$$

For a measurable set $E \subseteq \Theta$, define

$$\eta_M(E) = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\theta^{(i)} \in E\} \right], \qquad d_M = \sup_E \left| \eta_M(E) - \pi_M(E) \right|.$$

Thus η_M is the law of a uniformly chosen terminal particle, averaged over SMC randomness, and d_M is its total variation distance from π_M.

For $\zeta > 0$, assume the reference assigns positive mass to a near-optimal region:

$$a_\zeta = \nu_0(\{\theta : J(\theta) \ge J^\star - \zeta/2\}) > 0.$$

Let

$$\theta_{\text{term}} \in \operatorname*{arg\,max}_{\theta\in\{\theta^{(1)},\dots,\theta^{(N)}\}} \widehat{J}_M(\theta).$$

Then

$$\Pr(J^\star - J(\theta_{\text{term}}) > 2\epsilon_J + \zeta) \le d_M + \frac{\exp(-\tau_M \zeta/2)}{a_\zeta}. \tag{17}$$

The same bound holds for a uniformly chosen terminal particle.

Proof. Define the set of policies with low estimated reward by

$$H_\zeta = \left\{\theta : \widehat{J}_M(\theta) < J^\star - \epsilon_J - \zeta\right\}.$$

On the near-optimal region defining a_ζ, (16) gives

$$\widehat{J}_M(\theta) \geq J^\star - \zeta/2 - \epsilon_J.$$

Since $\pi_M(\theta) \propto \nu_0(\theta)\exp(\tau_M \widehat{J}_M(\theta))$, bounding its numerator and normalizing constant yields

$$\pi_M(H_\zeta) \leq \frac{\exp(\tau_M(J^\star - \epsilon_J - \zeta))}{a_\zeta \exp(\tau_M(J^\star - \epsilon_J - \zeta/2))} = \frac{\exp(-\tau_M\zeta/2)}{a_\zeta}.$$

If $J^\star - J(\theta_{\mathrm{term}}) > 2\epsilon_J + \zeta$, then the uniform estimation bound implies

$$\widehat{J}_M(\theta_{\mathrm{term}}) < J^\star - \epsilon_J - \zeta.$$

Because θ_{term} maximizes the estimated reward within the terminal population, every terminal particle must then belong to H_ζ. Consequently,

$$\begin{aligned}
&\Pr(J^\star - J(\theta_{\mathrm{term}}) > 2\epsilon_J + \zeta) \\
&\quad \leq \Pr\Big(\theta^{(i)} \in H_\zeta \text{ for all } i\Big) \\
&\quad \leq \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\theta^{(i)} \in H_\zeta\}\right] \\
&\quad = \eta_M(H_\zeta) \\
&\quad \leq \pi_M(H_\zeta) + d_M.
\end{aligned}$$

Combining these inequalities proves the result without assuming independent particles. For a uniformly chosen terminal particle, the reward-error event implies membership in H_ζ directly, giving the same bound. □

Interpretation and scope. The bound separates reward-estimation error ϵ_J, target concentration through τ_M and a_ζ, and particle-approximation error d_M. A small a_ζ means that the reference assigns little mass to near-optimal policies. For fixed ζ and a_ζ, increasing τ_M reduces the concentration term, although it may also make sampling more difficult and affect d_M. The bound holds at finite τ_M and does not require a zero-temperature limit.

Adaptive SMC convergence can be established under suitable assumptions (Beskos et al., 2016). Obtaining an explicit rate for d_M here requires a separate analysis of the adopted adaptation, mutation, and resampling rules. In particular, results for multinomial resampling do not automatically establish the required total-variation bound for systematic resampling. MH invariance alone does not provide such a rate.

If (16) holds with failure probability δ_J, and $d_M \leq \bar{d}$ whenever it holds, the unconditional bound becomes

$$\Pr(J^\star - J(\theta_{\mathrm{term}}) > 2\epsilon_J + \zeta) \leq \delta_J + \bar{d} + \frac{\exp(-\tau_M\zeta/2)}{a_\zeta},$$

for fixed J, ν_0, and τ_M. With the preservation penalty enabled, the same argument applies to the population penalized objective R and estimate $\widehat{R}_M$, provided their uniform estimation error is bounded.

The policy θ_{term} is selected from the terminal population using the final search reward. The deployed policy $\theta^\star$ in (12) is selected from an archive-based shortlist using validation scores. Applying the bound to that additional selection step requires corresponding validation and candidate-retention assumptions.

D Details of Steering Methods

All quantities attached to a layer are indexed by l or l_j; an activation coordinate is indexed by a. Directions, masks, and update constructors are computed from the permitted prototype split and do not use held-out test data. The descriptions below define the four operators used in our experiments and their local parameter blocks. Task-specific search ranges and budgets are separate experimental choices.

D.1 Additive Methods

Contrastive Activation Addition. CAA constructs a steering direction by averaging activation differences between paired positive and negative examples (Rimsky et al., 2024). Let v_l^{CAA} denote the resulting fixed contrastive direction at layer l. For the local parameter block

$$\theta_j^{\mathrm{CAA}} = \alpha_j,$$

the additive update and intervention are

$$\Delta_{\mathrm{CAA},l_j}(h;\theta_j^{\mathrm{CAA}}) = \alpha_j v_{l_j}^{\mathrm{CAA}}, \qquad h'_{l_j} = h_{l_j} + \alpha_j v_{l_j}^{\mathrm{CAA}}. \tag{18}$$

The direction is fixed after construction, while the SMC policy determines the intervention layers and their layer-local coefficients. Because the same direction is applied to every input at a selected layer, the update does not otherwise depend on h.

COLD-Steer. COLD-Steer approximates the activation change induced by one step of learning on in-context examples (Sharma & Trivedi, 2026). Let

$$\widetilde{\Delta}_{\mathrm{COLD},l}(h;\mathcal{D}_{\mathrm{proto}})$$

denote the estimated unscaled change at layer l. For the local parameter block

$$\theta_j^{\mathrm{COLD}} = \alpha_j,$$

the complete additive update and intervention are

$$\begin{aligned} \Delta_{\mathrm{COLD},l_j}(h;\theta_j^{\mathrm{COLD}}) &= \alpha_j \widetilde{\Delta}_{\mathrm{COLD},l_j}(h;\mathcal{D}_{\mathrm{proto}}), \\ h'_{l_j} &= h_{l_j} + \alpha_j \widetilde{\Delta}_{\mathrm{COLD},l_j}(h_{l_j};\mathcal{D}_{\mathrm{proto}}). \end{aligned} \tag{19}$$

The kernel or finite-difference construction is part of the operator definition. If that construction is fixed, the local search block contains only α_j; any additional tuned construction parameters must be included explicitly in θ_j^{COLD}. The update may change both direction and norm. Constructing it can require gradients or auxiliary computations, even though the outer SMC search uses only scalar scores.

AUSteer. AUSteer ranks activation units using activation-momentum statistics and applies input-adaptive sparse rescaling (Feng et al., 2026). Let $B_{l,k} \in \{0,1\}^{d_l}$ select k units at a fixed intervention site, and let $\gamma_l(h;\alpha)$ denote their strength vector. For the local parameter block

$$\theta_j^{\mathrm{AUSteer}} = (\log \alpha_j, k_j), \qquad \alpha_j > 0, \quad k_j \geq 1,$$

the additive update is

$$\Delta_{\mathrm{AUSteer},l_j}(h;\theta_j^{\mathrm{AUSteer}}) = B_{l_j,k_j} \odot \big(\gamma_{l_j}(h;\alpha_j) \odot h\big), \tag{20}$$

and the local intervention is

$$h'_{l_j} = h_{l_j} + B_{l_j,k_j} \odot \big(\gamma_{l_j}(h_{l_j};\alpha_j) \odot h_{l_j}\big). \tag{21}$$

For a selected coordinate a, this is

$$h'_{l_j,a} = \big(1 + \gamma_{l_j,a}(h_{l_j};\alpha_j)\big) h_{l_j,a}.$$

AUSteer therefore belongs to the additive-update family in (4), although its implementation is coordinatewise scaling rather than addition of a fixed contrastive vector. A layerwise policy may additionally enforce

$$\sum_j k_j \leq k_{\mathrm{budget}}.$$

Allocating units separately prevents a global top-k operation from assigning zero units to a nominally selected layer. Selected support alone, however, does not establish a nonzero behavioral effect.

Method	Family	Local control and steering signal
CAA	Additive	Strength of a fixed contrastive direction.
COLD-Steer	Additive	Strength of an estimated learning update.
AUSteer	Additive	Strength and count of selected activation units.
Spherical Steering	Rotational	Strength and gate for rotation toward a prototype.

Table 12: Operator-specific parameter blocks used in our experiments. All four methods share the same layerwise policy representation and outer SMC update.

D.2 Rotational Methods

Spherical Steering. For layer l, let

$$\mu_l = \frac{m_l^+ - m_l^-}{\|m_l^+ - m_l^-\|_2}, \qquad \hat{h} = \frac{h}{\|h\|_2}, \tag{22}$$

where m_l^+ and m_l^- are positive and negative prototype means. With antipodal prototypes and a common concentration κ, the vMF class-probability difference simplifies to

$$\delta_l(h) = -\tanh(\kappa \mu_l^\top \hat{h}), \qquad t_j(h) = \begin{cases} 0, & \delta_{l_j}(h) \le \beta_j, \\ \mathrm{clip}\left(\alpha_j \dfrac{\delta_{l_j}(h) - \beta_j}{1 - \beta_j}, 0, 1\right), & \delta_{l_j}(h) > \beta_j, \end{cases} \tag{23}$$

where $\beta_j < 1$ and κ is fixed unless explicitly searched (You et al., 2026). The local parameter block is

$$\theta_j^{\mathrm{Spherical}} = (\alpha_j, \beta_j),$$

and the intervention is

$$h'_{l_j} = \|h_{l_j}\|_2 \,\mathrm{Slerp}\big(\hat{h}_{l_j}, \mu_{l_j}, t_j(h_{l_j})\big). \tag{24}$$

For unit vectors u, v and $\varphi = \arccos(u^\top v) \in (0, \pi)$,

$$\mathrm{Slerp}(u, v, t) = \frac{\sin((1-t)\varphi)}{\sin\varphi} u + \frac{\sin(t\varphi)}{\sin\varphi} v. \tag{25}$$

The realized rotation is $t_j(h)\varphi_{l_j}(h)$, where

$$\varphi_l(h) = \arccos(\hat{h}^\top \mu_l).$$

Near coincident directions, a stable limit is needed; antipodal directions require a specified choice of geodesic. The gate can make a selected layer inactive. Activation rates and layer-removal ablations are therefore useful when interpreting a K-layer policy.

D.3 Geometry and Its Limits

For any additive update Δ, the change in squared norm is

$$\|h + \Delta\|_2^2 - \|h\|_2^2 = 2\langle h, \Delta\rangle + \|\Delta\|_2^2. \tag{26}$$

It may be positive or negative; additive steering does not always increase the activation norm. A pointwise orthogonal update satisfies

$$\|Q(h)h\|_2 = \|h\|_2,$$

but later nonlinear layers can still change both activation norms and the output distribution. Neither geometry by itself guarantees downstream quality.

For a fixed input and a differentiable map $z(h)$ from the intervention site to logits, let

$$p = \mathrm{softmax}(z(h)) \qquad \text{and} \qquad A(h) = \frac{\partial z}{\partial h}.$$

A small perturbation Δh gives the local expansion

$$D_{\mathrm{KL}}\big(p(h+\Delta h)\,\|\,p(h)\big) = \frac{1}{2}\Delta h^\top A(h)^\top\big(\mathrm{diag}(p) - pp^\top\big)A(h)\Delta h + o(\|\Delta h\|_2^2). \tag{27}$$

This follows by expanding log-sum-exp to second order. The sensitivity depends on the downstream Jacobian and the perturbation direction, not only its radial component. The formula concerns one fixed conditioning context; it is not a bound on a complete autoregressive or diffusion rollout. It motivates measuring preservation using the task evaluator instead of assuming it from norm control.

For spherical interpolation with $u = \hat{h}$ and target μ, define

$$e = \frac{\mu - \cos\varphi\, u}{\sin\varphi}$$

when $0 < \varphi < \pi$. Then

$$\mathrm{Slerp}(u, \mu, t) = \cos(t\varphi)u + \sin(t\varphi)e. \tag{28}$$

The update is a rotation through $t\varphi$ in $\mathrm{span}(u, \mu)$. For a small offset η in this plane,

$$\|h\|_2\big(\cos\eta\, u + \sin\eta\, e\big) = h + \eta\|h\|_2 e - \frac{1}{2}\eta^2 h + O(\|h\|_2|\eta|^3). \tag{29}$$

The first-order term is tangential, and the second-order term enforces norm preservation. Because the spherical interpolation plane depends on both the current activation and its target, different inputs need not be rotated in a common plane. Across transformer layers, even fixed additive interventions interact through the intervening nonlinear computation, which supports joint layerwise optimization for both operator families.

E Multimodality across Layers

Jointly steering two layers produces a search landscape that is multimodal across both the selected layer pair and the associated hyperparameters. To examine this structure, we collect the complete particle population from the final iteration of ActivationSMC and plot its empirical distribution over the two-layer Spherical Steering policies. As shown in Figure 7, the particles do not concentrate around a single layer–parameter configuration. Instead, they occupy multiple separated regions of the policy space, indicating that distinct combinations of intervention layers and steering parameters achieve competitive rewards. This multimodality highlights the limitations of optimizing one layer or hyperparameter at a time, since committing to a single configuration can exclude other promising modes.

Within a fixed layer configuration, the objective landscape over α, β, and ℓ is also highly irregular and nonconvex. Although these continuous parameter slices are not always cleanly multimodal, they contain distinct peaks and uneven regions that can trap hyperparameter optimization.

To visualize this irregular landscape in terms of accuracy, and where each method lands within it, Figure 8 shows 2D slices of the full (layer, α, β) grid search on R-Bench/LLaVA, with the Native and ActivationSMC solutions marked directly on the accuracy surface. These slices reveal a highly irregular landscape with several locally favorable regions (also visualized against the final particle distribution in Figure 6). Native's fixed configuration lands in a comparatively low-accuracy region, whereas ActivationSMC converges to a distinct high-accuracy region.

F Classical Convergence Guarantees

Let $\pi_m^N = N^{-1}\sum_{i=1}^N \delta_{\theta_i^{(m)}}$ denote the particle population after the m-th reweight–resample–MH cycle. Suppose the reward is bounded, say $r \in [0, 1]$, so that each potential satisfies $1 \le G_m \le e^{\tau_m - \tau_{m-1}}$. Classical SMC theory (Del Moral, 2004; 2013) then gives, for every $p \ge 1$ and bounded test function φ,

$$\sqrt{N}\,\mathbb{E}\big[\big|\pi_m^N(\varphi) - \pi_m(\varphi)\big|^p\big]^{1/p} \le a_p\,\|\varphi\|_\infty \sum_{q=0}^{m} \kappa_{q,m}\,\beta_{q,m}. \tag{30}$$

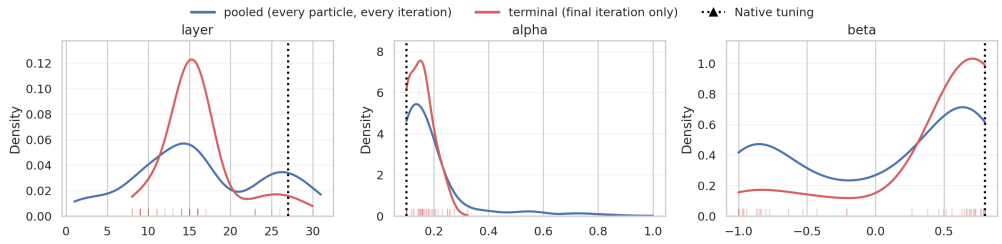


Figure 6: Hyperparameter particle distribution of Spherical Steering on R-Bench for one intervention layer. The terminal population retains particles in several separated high-reward regions. By coupling layer and parameter optimization, the population avoids premature commitment to a single suboptimal mode in this example. The native optimization falls for the lower reward mode.

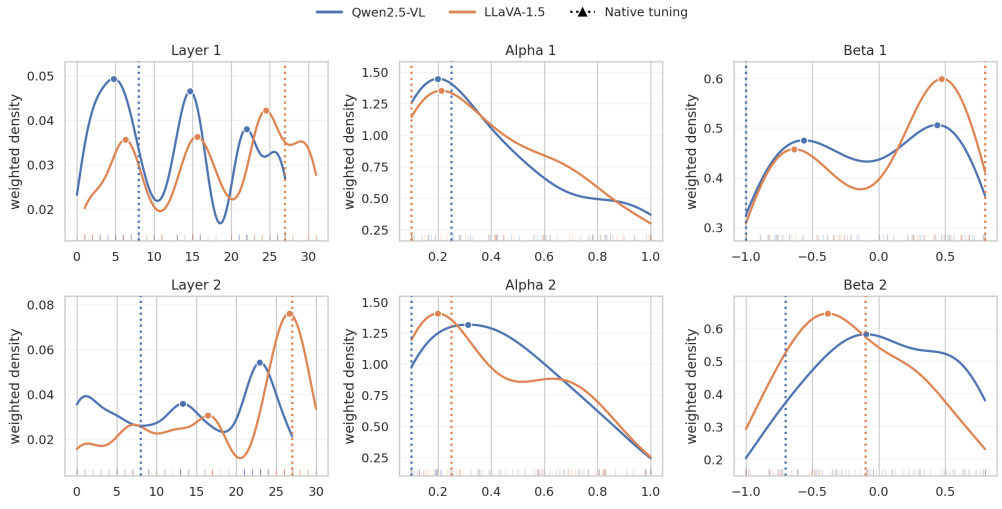


Figure 7: Marginal empirical distribution of the particles from the final ActivationSMC iteration for two-layer Spherical Steering on R-Bench, shown for Qwen and LLaVA. The presence of multiple separated concentrations across layer choices and steering parameters indicates that several distinct policies remain competitive at the end of the search.

Here the constant a_p depends only on p. $\beta_{q,m}$ is the Dobrushin contraction coefficient of the normalized Feynman–Kac semigroup that propagates the population from step q to step m, combining reweighting by G with the MH moves, and $\kappa_{q,m}$ bounds the relative oscillation of the accumulated potentials. The bound shows that SMC attains the Monte Carlo rate $O(N^{-1/2})$, and that well-mixing MH kernels make $\beta_{q,m}$ small, so early errors are forgotten. If the MH proposals draw from ν_0 with a fixed p_g probability, $\beta_{q,m}$ decays geometrically in $m - q$, and the right-hand side of Eq. (30) is bounded uniformly in m by a constant depending on p, p_g, and τ_M (Del Moral, 2004, Ch. 7).

Under the same conditions, the particle estimates satisfy a central limit theorem, $\sqrt{N}\big(\pi_m^N(\varphi) - \pi_m(\varphi)\big) \Rightarrow \mathcal{N}\big(0, \sigma_m^2(\varphi)\big)$ (Chopin, 2004; Del Moral, 2004), and their bias is of smaller order, $\big|\mathbb{E}[\pi_m^N(\varphi)] - \pi_m(\varphi)\big| \le c_m \|\varphi\|_\infty / N$ (Del Moral, 2004; Del Moral et al., 2009), so the marginal law of each particle converges to π_m at rate $1/N$. Beskos et al. (2016) establish consistency and the central limit theorem when the tempering increments are chosen adaptively by an ESS rule such as Eq. (10). These results are proved for multinomial resampling (Del Moral, 2004; Chopin, 2004; Douc & Moulines, 2008); see Douc et al. (2005) for a comparison of resampling schemes. The systematic resampling used in practice lacks a general central limit theorem (Gerber et al., 2019).

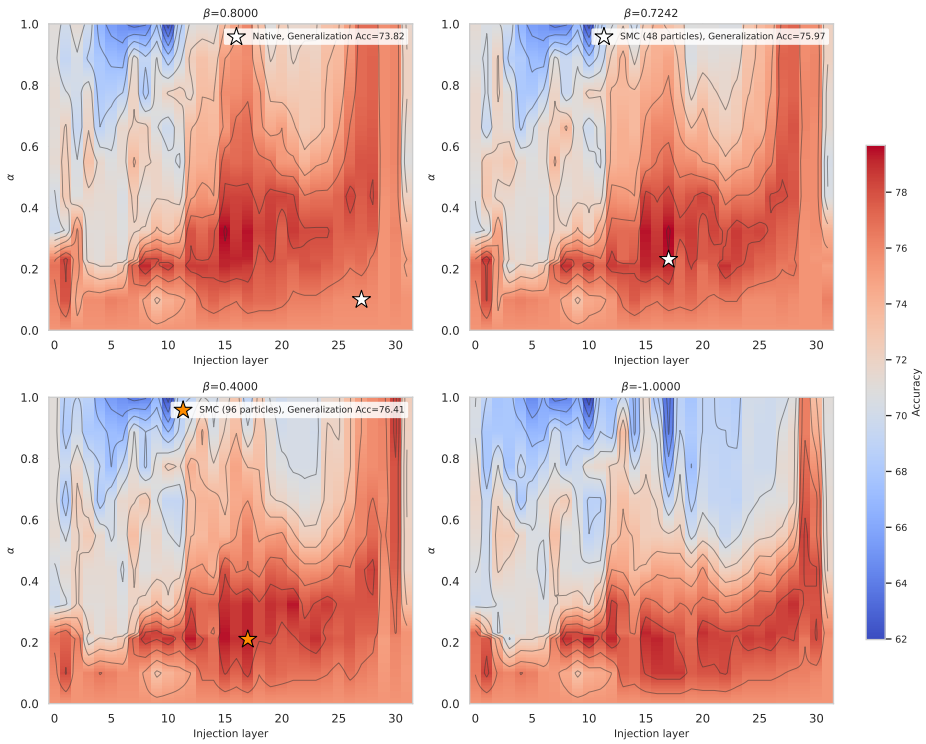


Figure 8: Layer $\times$ α slice (faceted by β) of the full three-dimensional (layer, α, β) grid search landscape for Spherical Steering on R-Bench/LLaVA, with Native and ActivationSMC's selected configurations marked. Native settles in a lower-reward region of the landscape, while SMC converges into a higher-reward mode, which corresponds to better held-out generalization accuracy for both particle counts.