GMPROP: A LEARNED IMPORTANCE-SAMPLING PROPOSAL FOR PRF ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Softmax attention is powerful but expensive: its cost grows quadratically with sequence length. Positive random features (PRF) can approximate softmax attention and reduce its computational cost to linear time. To date, most methods used a standard normal distribution to sample random features. However, such an approach may poorly reflect the query and key distribution of a pretrained Transformer and therefore require many features for good accuracy. In this work we introduce GMProp. GMProp learns a sampling distribution from which the PRF are sampled. It represents the sampling distribution as a small mixture of Gaussians learned offline from the model's query and key representations. At inference time, no fine-tuning or input-dependent optimization is required: the resulting method has the same linear-time attention computation as standard PRF attention and keeps its constant cost per generated token. Importance weights keep the kernel estimate unbiased for any learned proposal and any number of features. We provide theoretical justification for the design of GMProp and provide a thorough empirical investigation of the method. Across three pretrained language models, GMProp consistently improves over existing PRF baselines, and the learned proposals continue to perform well on a held-out corpus without being refit.

1 INTRODUCTION

Softmax attention costs $O(L^2)$ in the sequence length L. *Positive random features* (PRF) reduce the *prefill*, the pass over a prompt, to linear time and make each generated token cost the same whatever its position, by writing the attention kernel $\kappa(q,k) = e^{q^\top k}$ as an expectation of positive features and contracting in feature space, as in FAVOR+ (Choromanski et al., 2021; Rahimi & Recht, 2007): for $\omega \sim \mathcal{N}(0, I_d)$ and $\phi^+_\omega(x) = e^{\omega^\top x - \frac{1}{2}\|x\|^2}$, averaging m draws of $\phi^+_\omega(q)\phi^+_\omega(k)$ is unbiased for $e^{q^\top k}$, namely, $\mathbb{E}_{\omega\sim\mathcal{N}(0,I_d)}\left[\phi^+_\omega(q)\phi^+_\omega(k)\right] = e^{q^\top k}$. Its finite-sample accuracy depends mainly on the law used to sample the features, and the default isotropic Gaussian ignores that pretrained queries and keys are large in a few shared directions, so replacing a pretrained model's exact attention by the PRF estimator, without updating its weights, often needs a large m.

Adapting the model to the approximation. Methods that make attention linear differ in what they are allowed to modify, and most of them modify the weights of the model they are applied to. FAVOR+ substituted into a pretrained model is inaccurate at small m. Quality is recovered by continuing to train the model with the feature map in place, which moves (W_Q, W_K, W_V), so the model no longer computes the attention it was trained with (Choromanski et al., 2021). DARKFormer trains the projections alongside a learned Σ and is unbiased for the modified kernel $e^{q^\top \Sigma k}$ rather than for $e^{q^\top k}$ (Farzam et al., 2026), LARA and EVA train query-adaptive proposals jointly with the model (Zheng et al., 2022; 2023), and the Coulomb particle model replaces the sampling law by m trained feature vectors, so the feature count becomes part of the trained object (Badiei Khuzani et al., 2026).

Adapting the sampling at inference. The remaining methods do not update the weights of the pretrained model and pay inside the forward pass instead. FAVOR++ and FAVOR# re-solve their feature-map parameters in closed form from the current queries and keys, once per forward pass (Likhosherstov et al., 2022; 2023). The solve is not a fixed overhead: it reads all L positions, so

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.

it adds $O(Ld)$ for FAVOR++ and $O(Ld^2 + md^2 + d^3)$ for FAVOR# to every prefill, and repeats at every decoded token (Table 5). It is the run-time cost of a proposal that depends on the current queries, and it gives up the constant cost per generated token. We instead fix one law of ω per head, fitted once offline and shared by every query, so nothing is re-solved and that cost stays constant.

Pair sums and the optimal sampling law. For a query q and a key k, the PRF estimate of $\kappa(q, k)$ averages $p_I(\omega)\phi^+_\omega(q)\phi^+_\omega(k)$ over the draws ω. Completing the square shows that this product is a Gaussian bump of unit width centred at $q + k$, which we call their *pair sum* (section 3.3), so the estimate is accurate only when enough draws land near $q + k$. In a pretrained model, queries and keys are large along a few shared directions, and a pair attends strongly when its query and key point the same way. The pair sums of the strongly attending pairs therefore lie along those directions, far from the origin, where standard Gaussian draws rarely land. Theorem 1 states this for all pairs at once: the sampling law with the smallest Monte Carlo variance is proportional to the square root of a Gaussian mixture with one component at each pair sum, weighted by the square of that pair's kernel value. It cannot be sampled, since its normalizing constant has no closed form (section 3.6).

GMProp: a learned Gaussian mixture. **GMProp** (Gaussian Mixture Proposal) draws the features from a mixture of a few Gaussians ψ instead of the standard Gaussian, learned so that its components cover the pair sums, and corrects each draw with the importance weight p_I/ψ. A Gaussian mixture is the natural tractable stand-in for the optimal law: it has the same form, with a few components in place of one per pair, and Theorem 2 shows that the family contains a law close to the optimum. It is also easy to sample and to train. A draw is $\omega = \mu_c + \sigma_c z$, where c indexes a component drawn with probability equal to its mixing weight, μ_c and σ_c are that component's mean and width, and $z \sim \mathcal{N}(0, I_d)$. Thus the gradient of any loss, including the model's own next-token loss, can reach the mixture's parameters through the estimator (section 3.5). GMProp thus leaves the model exact, the estimator unbiased for the kernel and the sampling law fixed at inference.

2 PRELIMINARIES

Softmax attention. Let $[n] := \{1, \dots, n\}$. An attention head maps the L positions of a sequence, with queries, keys and values $q_i, k_i, v_i \in \mathbb{R}^d$ for $i \in [L]$, through the kernel $\kappa(q, k) = e^{q^\top k}$ to

$$y_i = \frac{\sum_j \kappa(q_i, k_j)\, v_j}{\sum_j \kappa(q_i, k_j)}, \qquad \text{i.e.} \qquad Y = D^{-1}KV, \quad K_{ij} = \kappa(q_i, k_j), \quad D = \operatorname{diag}(K\mathbf{1}), \tag{1}$$

the sums running over the positions a query may attend to, that is $j \le i$ under a causal mask. Here d is the head dimension and the usual $1/\sqrt{d}$ scaling of the dot product is absorbed into q and k once and for all, so it appears nowhere below. $K \in \mathbb{R}^{L\times L}$ is the *kernel matrix*, formed at cost $O(L^2 d)$, and keys appear only as the vectors k_j. The *captured* dataset $\mathcal{D} = \{(q_i, k_i)\}_{i\in[N]}$ holds, in token order, the query and key of each of N tokens from several contexts, recorded in forward passes of the pretrained model. The *pair law* $\mathcal{P}$ puts *pair weight* $\mathcal{P}_{ij}$ on the pair (q_i, k_j), $i, j \in [N]$, of $\mathcal{D}$: by default uniform over the causal pairs, $j \le i$ within one context, and zero across contexts, and in general any weighting. We write $\mathbb{E}_{q,k}$ for the average over $(q, k) \sim \mathcal{P}$, so $\mathbb{E}_{q,k}[f] = \sum_{i,j} \mathcal{P}_{ij} f(q_i, k_j)$. A transformer layer with input rows $x_1, \dots, x_L \in \mathbb{R}^{d_{\text{model}}}$ runs H such heads in parallel (Vaswani et al., 2017). Head h owns three projections $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in \mathbb{R}^{d_{\text{model}}\times d}$, forms $q_i = W_Q^{(h)\top} x_i$, $k_j = W_K^{(h)\top} x_j$ and $v_j = W_V^{(h)\top} x_j$, and returns $Y^{(h)}$ by equation 1. The layer's output is $[\, Y^{(1)} \cdots Y^{(H)} \,] W_O$ with $W_O \in \mathbb{R}^{Hd\times d_{\text{model}}}$.

PRF attention. Let $p_I : \mathbb{R}^d \to (0, \infty)$, $p_I(\omega) = (2\pi)^{-d/2} e^{-\|\omega\|^2/2}$, be the density of the standard Gaussian $\mathcal{N}(0, I_d)$, the *reference* distribution of this paper. FAVOR+ (Choromanski et al., 2021) rests on the PRF identity: for $\omega, x \in \mathbb{R}^d$,

$$\phi^+_\omega(x) = e^{\omega^\top x - \frac{1}{2}\|x\|^2}, \qquad e^{q^\top k} = \mathbb{E}_{\omega\sim\mathcal{N}(0,I_d)}\big[\phi^+_\omega(q)\, \phi^+_\omega(k)\big], \tag{2}$$

which follows from the Gaussian moment-generating function $\mathbb{E}_{\omega\sim\mathcal{N}(0,I_d)}[e^{\omega^\top a}] = e^{\|a\|^2/2}$ at $a = q + k$. Drawing $\omega_1, \dots, \omega_m \sim \mathcal{N}(0, I_d)$ i.i.d. and forming $\Phi_Q, \Phi_K \in \mathbb{R}^{L\times m}$ with entries $(\Phi_Q)_{is} =$

$\phi^+_{\omega_s}(q_i)/\sqrt{m}$ and $(\Phi_K)_{js} = \phi^+_{\omega_s}(k_j)/\sqrt{m}$ gives $\mathbb{E}_{\omega_{1:m}\sim\mathcal{N}(0,I_d)}[\Phi_Q\Phi_K^\top] = K$, the draws being the only random quantities, and attention is estimated by

$$\widehat{Y} = \widehat{D}^{-1}\,\Phi_Q\big(\Phi_K^\top V\big), \qquad \widehat{D} = \operatorname{diag}\big(\Phi_Q(\Phi_K^\top \mathbf{1})\big), \tag{3}$$

evaluated right to left in $O(Lmd)$ time and $O(md)$ memory beyond the inputs, without forming an $L \times L$ matrix. $\widehat{W} = \widehat{D}^{-1}\Phi_Q\Phi_K^\top$ is the estimated attention matrix, so $\widehat{Y} = \widehat{W}V$. Under a causal mask the products become prefix sums. At decoding, each new token adds its key to these sums, so it costs $O(md)$ time and memory whatever its position, against $O(Ld)$ time and an $O(Ld)$ key-value (KV) cache for softmax attention. This paper changes only the law the ω_s are drawn from, and equation 3 is used as is.

Pathwise gradients. For a family of probability densities $r_\theta : \mathbb{R}^d \to (0,\infty)$ with parameters θ and a function $f(\theta,\omega) \in \mathbb{R}$, the gradient of $\mathbb{E}_{\omega\sim r_\theta}[f(\theta,\omega)]$ with respect to θ is taken *pathwise* (Mohamed et al., 2020; Kingma & Welling, 2014; Rezende et al., 2014): the draw is written as a deterministic, differentiable map $\omega = \omega_\theta(z)$ of noise z whose law does not depend on θ, such that drawing z and computing $\omega_\theta(z)$ is the same as drawing ω from r_θ. Differentiating through θ is possible according to the following identity:

$$\nabla_\theta\, \mathbb{E}_{\omega\sim r_\theta}\big[f(\theta,\omega)\big] \;=\; \mathbb{E}_z\big[\nabla_\theta f\big(\theta,\omega_\theta(z)\big)\big], \tag{4}$$

where ∇_θ acts on both arguments of f. A discrete choice in the draw whose law depends on θ, such as the component index of a mixture, is *detached*: its sampled value is treated as a constant and no gradient flows through the categorical draw (Goodfellow et al., 2016, §20.9), which leaves the gradient exact in the parameters of the map and gives no gradient to the probabilities of the discrete choice through the draw.

3 METHOD

3.1 THE ESTIMATOR

FAVOR+ estimates the kernel identity equation 2 by averaging m draws from the reference distribution $\mathcal{N}(0, I_d)$. We instead draw from a learned *proposal*, a probability density $r_\theta : \mathbb{R}^d \to (0,\infty)$ with $\int r_\theta(\omega)\,\mathrm{d}\omega = 1$ and $r_\theta(\omega) > 0$ for all ω. We then estimate the kernel with the importance-sampling correction weight $w(\omega) = p_I(\omega)/r_\theta(\omega)$:

$$\hat{\kappa}_\theta(q,k) = \frac{1}{m}\sum_{j=1}^{m} w(\omega_j)\,\phi^+_{\omega_j}(q)\,\phi^+_{\omega_j}(k), \qquad \omega_j \sim r_\theta. \tag{5}$$

Each term satisfies $\mathbb{E}_{\omega\sim r_\theta}[w(\omega)\phi^+_\omega(q)\phi^+_\omega(k)] = \mathbb{E}_{\omega\sim\mathcal{N}(0,I_d)}[\phi^+_\omega(q)\phi^+_\omega(k)] = \kappa(q,k)$ by equation 2, so $\mathbb{E}_{\omega_{1:m}\sim r_\theta}[\hat{\kappa}_\theta(q,k)] = \kappa(q,k)$ for every θ (Proposition 1, Appendix A). For queries q_i and keys k_j we write $\widehat{K}_{ij} = \hat{\kappa}_\theta(q_i,k_j)$ for the estimate of the kernel matrix K of equation 1. The linear-attention computation equation 3 is unchanged: Φ_Q and Φ_K are built from the draws $\omega_j \sim r_\theta$ of equation 5, with the weights $w(\omega_j)$ folded into the features, so attention stays linear in the sequence length at the same deployed cost.

3.2 GMPROP: THE FAMILY

GMProp uses as its proposal a mixture of C Gaussians on $\mathbb{R}^d$, the space of the feature draws ω (Bishop, 2006),

$$r_\theta(\omega) = \sum_{c=1}^{C} \lambda_c\,\mathcal{N}\big(\omega \mid \mu_c,\ \sigma_c^2 I_d\big), \qquad \lambda = \operatorname{softmax}(\nu), \tag{6}$$

with learned *component means* $\mu_c \in \mathbb{R}^d$, scalar *widths* σ_c and *mixing coefficients* $\lambda_c = \operatorname{softmax}(\nu)_c$. The stacks over components are $\mu = (\mu_1,\dots,\mu_C)$, σ and ν, and the parameters are

$$\theta = \big(\underbrace{\mu}_{\text{component means}}, \underbrace{\log\sigma}_{\text{widths}}, \underbrace{\nu}_{\text{mixing coefficients}}\big), \qquad \mu \in \mathbb{R}^{C\times d},\ \nu \in \mathbb{R}^C, \tag{7}$$

and $\log\sigma \in \mathbb{R}^C$, one width per component, for $Cd + 2C$ floats.

3.3 THE INPUT GAUGE

The problem. For one query–key pair, the integrand of the PRF identity equation 2 is a Gaussian bump, as completing the square shows,

$$p_I(\omega)\,\phi_\omega^+(q)\,\phi_\omega^+(k) = e^{q^\top k}\,\mathcal{N}(\omega \mid q+k,\, I_d), \tag{8}$$

of unit width, centred at the *pair sum* $q+k$ and weighted by the kernel value. A proposal therefore estimates a pair's kernel with low variance when it puts mass near that pair's sum, so r_θ has to cover the pair sums $q_i + k_j$ of the pairs with large kernel values. The components of equation 6 are spherical, so each covers a ball. In a pretrained model the heavy pair sums instead form a long, thin streak. Pretrained queries and keys are large along a few shared directions (Ethayarajh, 2019; Timkey & van Schijndel, 2021; Sun et al., 2024), and a pair is heavy when its query and key point the same way, so the heavy sums line up along those directions. Covering a streak with balls takes many components, laid end to end.

The gauge. Besides r_θ, we learn the *input gauge*, an invertible $A \in \mathbb{R}^{d\times d}$ that changes the coordinates of the inputs,

$$\tilde{q} = Aq, \qquad \tilde{k} = A^{-\top}k, \qquad \text{so} \qquad \tilde{q}^\top\tilde{k} = q^\top A^\top A^{-\top}k = q^\top k. \tag{9}$$

The kernel $\kappa(q,k) = e^{q^\top k}$ depends on q and k only through $q^\top k$, so the kernel, the attention matrix and the model output are the same for every A, and the estimate equation 5 formed from $\tilde{q}$ and $\tilde{k}$ stays unbiased. What changes are the pair sums, which move from $q_i + k_j$ to $Aq_i + A^{-\top}k_j$. From here on, q and k denote the gauged inputs, with $A = I$ when the gauge is off.

Why it works. An invertible A can rotate the coordinates, stretch them along any set of directions and shear them, and each choice moves the pair sums differently. A is trained jointly with θ so that the gauged pair sums $Aq_i + A^{-\top}k_j$ spread about equally in every direction instead of forming a streak, and pair sums spread that way are covered by a few spherical components. One $d \times d$ matrix thus does the work of many extra components. A is kept well-conditioned during training, and at inference it folds into the cached projections ΩA and $\Omega A^{-\top}$, as Appendix C describes.

3.4 THE OBJECTIVE

The target. The target is the expected Monte Carlo variance of equation 5 (Farzam et al., 2026): the variance over the feature draws, averaged over the query–key pairs. For i.i.d. draws that variance is $1/m$ times the variance of the estimator at a single draw,

$$\hat{\kappa}_\theta(q,k;\omega) = w(\omega)\,\phi_\omega^+(q)\,\phi_\omega^+(k), \qquad \omega \sim r_\theta, \tag{10}$$

and the target is

$$\arg\min_\theta\ \mathbb{E}_{q,k}\big[\mathrm{Var}_{\omega\sim r_\theta}\big(\hat{\kappa}_\theta(q,k;\omega)\big)\big] = \arg\min_\theta\ \mathbb{E}_{q,k}\,\mathbb{E}_{\omega\sim r_\theta}\Big[\big(\hat{\kappa}_\theta(q,k;\omega) - \kappa(q,k)\big)^2\Big], \tag{11}$$

where θ enters $\hat{\kappa}_\theta(q,k;\omega)$ twice, as the law ω follows and as the weight $w = p_I/r_\theta$, while $\kappa(q,k)$ does not depend on θ. The equality holds because $\mathbb{E}_{\omega\sim r_\theta}[\hat{\kappa}_\theta(q,k;\omega)] = \kappa(q,k)$ at every θ (Proposition 1). The experiments couple the m draws (Appendix G), which leaves the estimate unbiased. Adaptive importance sampling (Ryu & Boyd, 2014; Lemaire & Pagès, 2010; Müller et al., 2019) minimizes the second moment $\mathbb{E}_{q,k}\mathbb{E}_{\omega\sim r_\theta}[\hat{\kappa}_\theta(q,k;\omega)^2]$, in place of this variance: the same minimizer, since $\mathbb{E}_{\omega\sim r_\theta}[\hat{\kappa}_\theta(q,k;\omega)^2] = \mathrm{Var}_{\omega\sim r_\theta}(\hat{\kappa}_\theta(q,k;\omega)) + \kappa(q,k)^2$ with $\kappa(q,k)$ free of θ, but noisier at finite m, since the exact $\kappa(q,k)$ in the squared error is a control variate (Appendix D).

The target underflows. The kernel values $e^{q^\top k}$ span hundreds of orders of magnitude, so all but a handful of pairs underflow to zero. Trained on equation 11 directly, the proposal does not move, and measured directly, the variance is set by those few pairs, so it reports the corpus's range rather than the estimator (Appendix E). We therefore train on, and report, surrogates.

Algorithm 1 GMProp training on a surrogate $\mathcal{L}$. Line 5 gives the law of one draw. Appendix G gives how the m draws are coupled and the step schedule of each surrogate.

Require: components C; features per step m; step size η

1: **Initialization** of each head (Algorithm 2): $\mu_c \leftarrow$ weighted k-means centres of the pair sums, $\lambda_c \leftarrow$ each cluster's mass, $\sigma_c \leftarrow 1$, $A \leftarrow I$ with the gauge off
2: **for** each step **do**
3: draw a minibatch of contexts
4: **for** each head and $s = 1, \dots, m$ **do**
5: $c_s \sim \mathrm{Cat}(\lambda)$, $z_s \sim \mathcal{N}(0, I_d)$, $\omega_s \leftarrow \mu_{c_s} + \sigma_{c_s} z_s$, $w_s \leftarrow p_I(\omega_s)/r_\theta(\omega_s)$ ▷ equation 14
6: **end for**
7: $\hat{\kappa}_\theta(q,k) \leftarrow \frac{1}{m}\sum_{s=1}^{m} w_s\, \phi^+_{\omega_s}(Aq)\, \phi^+_{\omega_s}(A^{-\top}k)$ for the causal pairs (q,k) of each head in the model's coordinates, read from $\mathcal{D}$ for $\mathcal{L}_{\log}$ and formed in the live forward for $\mathcal{L}_{\mathrm{CE}}$ ▷ equation 5
8: $\mathcal{L} \leftarrow$ equation 12 or equation 13 on the minibatch
9: $(\theta, A) \leftarrow (\theta, A) - \eta\, \mathrm{d}\mathcal{L}/\mathrm{d}(\theta, A)$ for each head ▷ equation 17, and equation 16 for $\mathcal{L}_{\log}$
10: **end for**
11: **return** θ and A of each head at the validation-best step

The surrogates. The primary surrogate objective we study in this work is the *log-kernel MSE*,

$$\mathcal{L}_{\log}(\theta) = \mathbb{E}_{q,k}\, \mathbb{E}_{\omega_{1:m}\sim r_\theta}\Big[\big(\log \hat{\kappa}_\theta(q,k) - \log \kappa(q,k)\big)^2\Big], \tag{12}$$

with $\hat{\kappa}_\theta$ the m-draw estimator equation 5 and $\log \kappa(q,k) = q^\top k$, a quantity of ordinary size and a reference the optimizer cannot move (Table 1). It keeps the target: a first-order expansion of $\log \hat{\kappa}_\theta(q,k)$ around $\hat{\kappa}_\theta(q,k) = \kappa(q,k)$ gives $\mathcal{L}_{\log}(\theta) \approx \mathbb{E}_{q,k}\big[\mathrm{Var}_{\omega_{1:m}\sim r_\theta}(\hat{\kappa}_\theta(q,k))/\kappa(q,k)^2\big]$, the *relative* variance, exact only as $\hat{\kappa}_\theta(q,k) \to \kappa(q,k)$ (Appendix E, equations 48–49). The secondary surrogate is the model's next-token cross-entropy. It scores the proposals by the model's own task, predicting the next token of the data:

$$\mathcal{L}_{\mathrm{CE}}(\theta) = \mathbb{E}_{x_{1:L}}\, \mathbb{E}_{\omega_{1:m}\sim r_\theta}\Big[-\frac{1}{L}\sum_{t=1}^{L} \log \hat{p}_\theta(x_t \mid x_{<t})\Big], \tag{13}$$

where $x_{1:L}$ is a context of L tokens and $\hat{p}_\theta$ is the model's next-token distribution in a live forward with $\hat{\kappa}_\theta$ in place of κ in every head. Here θ collects the mixture parameters of every head, and the inner expectation is over the m features each head draws from its own proposal, so all heads train at once. The model's weights are not updated.

3.5 Gradient estimation by pathwise gradients

Algorithm 1 states the procedure, and this subsection derives the gradient of its line 9. The surrogates of section 3.4 are functions of the estimator equation 5, whose m features are drawn from the mixture equation 6, so their gradient with respect to θ must pass through the draws.

Sampling. A draw from r_θ is made in two stages: an index $c \sim \mathrm{Cat}(\lambda)$ selects a component, and a standard Gaussian $z \sim \mathcal{N}(0, I_d)$ is mapped through

$$\omega = \omega_\theta(z; c) := \mu_c + \sigma_c z, \tag{14}$$

at $O(d)$ per draw. Given c, $\omega \sim \mathcal{N}(\mu_c, \sigma_c^2 I_d)$, and marginally over c it is a draw from the mixture r_θ itself, so the importance weight is $p_I(\omega)/r_\theta(\omega)$.

The gradient is taken pathwise by equation 4, with the index detached: the noise z does not depend on θ, while c does through $\lambda = \mathrm{softmax}(\nu)$, so the gradient is exact in μ and σ, and ν receives no gradient through the choice of the index. Writing each draw of equation 12 as $\omega_s = \omega_\theta(z_s; c_s)$, with $c_s \sim \mathrm{Cat}(\lambda)$ and $z_s \sim \mathcal{N}(0, I_d)$, turns the loss into an expectation over the index and the noise,

$$\mathcal{L}_{\log}(\theta) = \mathbb{E}_{q,k}\, \mathbb{E}_{c_{1:m}, z_{1:m}}\Big[\big(\log \hat{\kappa}_\theta(q,k) - \log \kappa(q,k)\big)^2\Big], \tag{15}$$

and equation 4 moves ∇_θ inside it. The same holds for $\mathcal{L}_{\mathrm{CE}}$ with the model's loss in place of the squared log-error.

With $\hat{\kappa}_\theta(q,k) = \frac{1}{m}\sum_{s=1}^{m} \hat{\kappa}_\theta(q,k;\omega_s)$, the chain rule gives

$$\frac{\mathrm{d}\mathcal{L}_{\log}}{\mathrm{d}\theta} = \mathbb{E}_{q,k}\,\mathbb{E}_{c_{1:m},z_{1:m}}\left[\frac{2\big(\log\hat{\kappa}_\theta(q,k) - q^\top k\big)}{\hat{\kappa}_\theta(q,k)}\;\frac{1}{m}\sum_{s=1}^{m}\frac{\mathrm{d}\,\hat{\kappa}_\theta(q,k;\omega_s)}{\mathrm{d}\theta}\right], \tag{16}$$

so the only derivative to take is that of a single draw. The gradient of $\mathcal{L}_{\mathrm{CE}}$ contains the same per-draw derivative, multiplied by the derivative of the model's loss with respect to $\hat{\kappa}_\theta(q,k)$, which the model's backward pass provides (Appendix G, equation 50). The one-draw estimator depends on θ through the weight p_I/r_θ and through ω_θ, so its derivative has two parts,

$$\frac{\mathrm{d}\,\hat{\kappa}_\theta(q,k;\omega)}{\mathrm{d}\theta} = \underbrace{\partial_\theta\hat{\kappa}_\theta(q,k;\omega)\big|_\omega}_{\text{through the weight}} + \underbrace{\big(\nabla_\omega\hat{\kappa}_\theta(q,k;\omega)\big)^\top\frac{\partial\omega_\theta}{\partial\theta}}_{\text{pathwise, through }\omega_\theta}. \tag{17}$$

At fixed ω the mixture parameters enter the estimator only through the weight p_I/r_θ, so the first part is $-\hat{\kappa}_\theta(q,k;\omega)\,\partial_\theta\log r_\theta$, with $\partial_\theta\log r_\theta$ in closed form in Appendix B. The second part needs $\partial\omega_\theta/\partial\theta$, which by equation 14 is I_d for the mean and $\sigma_c z$ for the log-width of the sampled component c, and zero for every other parameter, so only the first part reaches the mixing logits ν.

3.6 Why a mixture of Gaussians

This subsection finds the variance-optimal proposal over all densities and shows that a Gaussian mixture at the attention-weighted pair sums is its natural approximation. Sernau et al. (2024) derive the variance-optimal proposal of a PRF representation assuming that the two kernel arguments are independent. Their argument uses the assumption only through the decorrelation of the squared features across the pair, which the causal mask and any non-uniform weighting of the pairs break. We drop this assumption and state the optimum for an arbitrary weighting of the pairs.

Theorem 1 (The variance-optimal proposal, proved in Appendix A.1)**.** *Let $\mathcal{P}_{ij}$ be the pair weights of section 2. Over all densities $\psi > 0$ on $\mathbb{R}^d$, with $w = p_I/\psi$, the minimizer of the expected Monte Carlo variance,*

$$\psi^* = \arg\min_{\psi}\ \mathbb{E}_{q,k}\big[\mathrm{Var}_{\omega\sim\psi}\big(w(\omega)\,\phi^+_\omega(q)\,\phi^+_\omega(k)\big)\big],$$

is unique: it is proportional to the square root of a Gaussian mixture with one component at each pair sum,

$$\psi^*(\omega)\ \propto\ \sqrt{\sum_{i,j}\mathcal{P}_{ij}\,\kappa(q_i,k_j)^2\,\mathcal{N}\big(\omega\mid q_i+k_j,\tfrac{1}{2}I_d\big)}\,. \tag{18}$$

Theorem 1 gives the optimal proposal in closed form up to its normalizing constant, and that constant, the integral of the root of the mixture, has no closed form (Olsen & Hershey, 2007). Drawing from ψ^* therefore needs a numerical normalization or a rejection step, which rules it out as a sampler. We learn the Gaussian mixture equation 6 instead. Its C components are Gaussians, their means standing in for the many pair sums in equation 18, and it is preferred for three reasons. First, a draw is one component index and one Gaussian draw, as in line 5 of Algorithm 1. Second, its distance from ψ^* is controlled at every C by Theorem 2 in Appendix A.3. Third, its draws come from the map equation 14, so equation 4 gives low-variance pathwise gradients of any objective through the sampler, and no such map is available for ψ^*.

3.7 Initialization

The means μ_c, the widths σ_c and the mixing coefficients λ_c of equation 6 are initialized by weighted k-means on pair sums: draw t queries and t keys from $\mathcal{D}$, form the t^2 sums $x_{ij} = q_i + k_j$, give each the weight $w_{ij} \propto e^{q_i^\top k_j}$, normalized to sum to one over the t^2 pairs, and cluster these weighted points into C groups (Algorithm 2, Appendix G). Then $\mu_c = \sum_{(i,j)\in c} w_{ij}x_{ij}\big/\sum_{(i,j)\in c} w_{ij}$ is the weighted centre of cluster c, $\lambda_c = \sum_{(i,j)\in c} w_{ij}$ its total weight, and $\sigma_c = 1$. These choices follow Theorem 1: by equation 18, the optimal proposal is the root of a sum of squared unit-width Gaussian bumps, one at each pair sum $q_i + k_j$ with weight $\kappa(q_i,k_j)$, so its mass lies at the pair sums of the pairs with large kernel values. k-means minimizes quantization distortion rather than the Monte Carlo variance, so training then updates μ, σ and λ (section 3.5).

4 EXPERIMENTS

This section compares GMProp with the baselines on log-kernel approximation error, next-token cross-entropy, a cross-corpus test and cost.

Protocol. Models are `EleutherAI/pythia-70m` (d=64, 6 layers $\times$ 8 heads), `EleutherAI/pythia-410m` (d=64, 24 layers $\times$ 16 heads) and `allenai/OLMo-1B-hf` (d=128, 16 layers $\times$ 16 heads). Feature counts are $m \in \{16, 256, 4096\}$. Each proposal is fit once and scored at all three feature counts without retraining. Appendix G gives the settings.

Datasets. We fit and score on samples of each model's pretraining corpus (Appendix F), the Pile for Pythia (Gao et al., 2020; Biderman et al., 2023) and Dolma v1.5 for OLMo-1B (Soldaini et al., 2024; Groeneveld et al., 2024). Each holds 37.8M tokens in contexts of at most 512 tokens, split by document into 34.0M training, 1.9M validation and 1.9M test tokens. The UN Parallel Corpus sample of section 4.3 splits into 8.39M, 65,536 and 51,200. What θ has to cover is not text but the pair sums $\{q_i + k_j\}$, which depend on the model's projections, not on the corpus. The pretraining mixture is the broadest input distribution the model was trained on, so a proposal fit there covers the pair sums over that whole range, not over one corpus, as section 4.3 tests.

Baselines. We compare only methods that replace the feature-sampling law of a pretrained model without changing its weights. A method is admitted if it is unbiased for K_{ij}, can be scored at any feature count m from one checkpoint without retraining, and needs no trained feature map of its own. The admitted methods are the baselines of every table in this section: FAVOR+, FAVOR++, FAVOR#, simplex random features (SimRF) (Reid et al., 2023) and Coupled PRFs (Reid et al., 2025), the last two correlating the draws while keeping each one isotropic. FAVOR# is run as the symmetric variant its paper found strongest in every setting and used for its own transformer experiments, so it represents the other variants. As in that paper, it is re-solved in each forward pass from the live queries and keys, and it is configured as in Appendix G. Hybrid random features (Hybrid RFs) (Choromanski et al., 2022) qualify but are omitted, because their signed features make the sum that normalizes a query's attention weights zero or negative on most query rows, so those queries get no valid attention.

4.1 LOG-KERNEL APPROXIMATION ERROR

The GMProp rows of Table 1 are trained by Algorithm 1 on $\mathcal{L}_{\log}$, each head separately on its captured dataset $\mathcal{D}$, at m=128.

Table 1: Layer-0 log-kernel MSE (↓), per-head training at both gauge settings. Full table is Table 7 (Appendix G).

	Pythia-70M pile mix			Pythia-410M pile mix			OLMo-1B Dolma v1.5		
Method	16	256	4096	16	256	4096	16	256	4096
GMProp, with gauge	**69.4 ± 53.9**	**23.0 ± 20.6**	**9.14 ± 9.05**	**20.5 ± 13.1**	**5.18 ± 4.02**	**1.68 ± 1.50**	**29.4 ± 11.6**	**8.30 ± 3.79**	**2.97 ± 1.55**
GMProp, no gauge	79.4 ± 61.8	27.9 ± 24.9	11.4 ± 11.5	25.1 ± 15.6	6.48 ± 4.48	2.10 ± 1.63	33.6 ± 12.5	10.2 ± 4.37	3.78 ± 2.02
FAVOR#	217 ± 197	49.5 ± 43.5	14.1 ± 13.8	51.1 ± 46.8	9.32 ± 7.57	2.27 ± 2.00	54.3 ± 26.2	11.5 ± 5.49	3.46 ± 1.93
SimRF	$> 10^3$	$> 10^3$	$> 10^3$	124 ± 139	55.1 ± 72.2	22.5 ± 30.7	82.5 ± 29.6	30.2 ± 11.4	11.8 ± 5.85
FAVOR+	$> 10^3$	$> 10^3$	$> 10^3$	128 ± 139	56.6 ± 73.7	21.8 ± 29.6	83.4 ± 30.8	30.8 ± 12.2	11.9 ± 5.95
Coupled PRFs	$> 10^3$	$> 10^3$	$> 10^3$	128 ± 138	54.1 ± 72.7	21.9 ± 29.0	83.5 ± 30.4	30.2 ± 11.9	11.8 ± 5.90
FAVOR++	$> 10^3$	$> 10^3$	$> 10^3$	251 ± 327	101 ± 132	40.4 ± 58.7	122 ± 47.6	40.7 ± 15.2	15.9 ± 6.50

At m=256 the gauged proposal is 23.0 on Pythia-70M against 49.5 for FAVOR# and 1.76×10^3 for FAVOR+. The same ordering holds on Pythia-410M (5.18 against 9.32) and OLMo-1B (8.30 against 11.5). Learning the gauge helps in every cell here and in Table 2. One cell is worth naming: on OLMo-1B at m=4096, FAVOR# (3.46) surpasses the ungauged proposal (3.78), the only cell where a baseline surpasses a GMProp row, and loses to the gauged one (2.97).

4.2 NEXT-TOKEN CROSS-ENTROPY

The GMProp rows of Table 2 are trained by Algorithm 1 on $\mathcal{L}_{\mathrm{CE}}$, with every head at once on the contexts $x_{1:L}$ of equation 13, at m=256.

Table 2: Next-token CE (↓) and perplexity (↓) on each model's pretraining mixture. Full table is Table 8 (Appendix G).

	Pythia-70M pile mix			Pythia-410M pile mix			OLMo-1B Dolma v1.5		
Method	16	256	4096	16	256	4096	16	256	4096
Exact attention		3.10 (22.1)			2.34 (10.3)			2.46 (11.7)	
GMProp, with gauge	**5.37 (214)**	**5.07 (160)**	**4.91 (136)**	**5.01 (150)**	**4.51 (90.7)**	**4.31 (74.4)**	**5.55 (258)**	**5.08 (161)**	**4.75 (115)**
GMProp, no gauge	5.75 (313)	5.35 (210)	5.10 (163)	5.66 (286)	4.90 (135)	4.68 (108)	5.96 (388)	5.38 (217)	5.04 (155)
FAVOR#	6.78 (876)	6.16 (473)	5.61 (273)	7.93 ($> 10^3$)	6.73 (837)	5.70 (298)	8.36 ($> 10^3$)	7.22 ($> 10^3$)	6.49 (655)
SimRF	8.53 ($> 10^3$)	7.39 ($> 10^3$)	7.37 ($> 10^3$)	7.94 ($> 10^3$)	7.87 ($> 10^3$)	7.65 ($> 10^3$)	8.08 ($> 10^3$)	7.11 ($> 10^3$)	6.61 (742)
FAVOR+	7.76 ($> 10^3$)	7.58 ($> 10^3$)	7.30 ($> 10^3$)	8.28 ($> 10^3$)	7.85 ($> 10^3$)	7.69 ($> 10^3$)	8.62 ($> 10^3$)	6.98 ($> 10^3$)	6.45 (632)
Coupled PRFs	8.24 ($> 10^3$)	7.57 ($> 10^3$)	7.37 ($> 10^3$)	8.09 ($> 10^3$)	7.99 ($> 10^3$)	7.39 ($> 10^3$)	8.26 ($> 10^3$)	7.01 ($> 10^3$)	6.54 (689)
FAVOR++	9.57 ($> 10^3$)	8.57 ($> 10^3$)	8.03 ($> 10^3$)	8.91 ($> 10^3$)	8.86 ($> 10^3$)	8.39 ($> 10^3$)	8.55 ($> 10^3$)	7.22 ($> 10^3$)	6.80 (901)

Table 2 reports next-token CE with the model weights frozen. At m=256 the gauged proposal leads the best baseline by 1.09, 2.22 and 1.90 nats on the three models, and the gauge helps in every cell.

4.3 Cross-corpus test

The proposals of Table 1 and Table 2 were fit on each model's pretraining mixture. Tables 3 and 4 score them, with nothing refit, on 100 official United Nations documents from the UN Parallel Corpus v1.0 (UNPC; Ziemski et al., 2016), next to proposals trained on UNPC itself and the baselines run as published. The proposals trained on UNPC use the loss of the table they appear in, $\mathcal{L}_{\log}$ in Table 3 and $\mathcal{L}_{\mathrm{CE}}$ in Table 4. The corpus was chosen for its provenance. It is a named component of neither the Pile nor Dolma v1.5, and it is distributed as a structured parallel corpus — sentence-aligned XML files released for machine translation — rather than as the web text a crawl collects, so its documents are not expected to have contaminated either pretraining corpus or either fitting mixture. Appendix F states what this does and does not rule out.

Table 3: Layer-0 log-kernel MSE (↓) on the UN Parallel Corpus. The first two GMProp rows are trained on this corpus. The pretraining-mixture rows are the proposals of Table 1, scored unchanged. Full table is Table 9 (Appendix G).

	Pythia-70M			Pythia-410M			OLMo-1B		
Method	16	256	4096	16	256	4096	16	256	4096
GMProp, with gauge	**65.5 ± 53.1**	**21.8 ± 19.9**	**8.83 ± 8.92**	**20.6 ± 13.8**	**5.34 ± 4.15**	**1.74 ± 1.52**	**27.5 ± 9.71**	**7.77 ± 3.13**	**2.72 ± 1.23**
GMProp, no gauge	83.0 ± 64.0	28.3 ± 24.7	11.5 ± 10.9	25.7 ± 15.5	6.91 ± 5.06	2.32 ± 1.92	34.2 ± 10.9	10.3 ± 4.15	3.83 ± 1.92
GMProp, with gauge, pretraining mixture	71.8 ± 55.8	24.1 ± 21.9	9.78 ± 9.81	22.4 ± 14.1	5.81 ± 4.49	1.91 ± 1.69	31.9 ± 10.4	9.27 ± 3.66	3.38 ± 1.53
GMProp, no gauge, pretraining mixture	84.1 ± 65.8	29.9 ± 26.6	12.3 ± 12.3	26.8 ± 16.0	7.19 ± 5.03	2.39 ± 1.88	37.8 ± 12.3	11.4 ± 4.62	4.32 ± 2.22
FAVOR#	250 ± 223	57.2 ± 48.2	16.5 ± 15.6	61.2 ± 55.4	11.5 ± 9.07	2.87 ± 2.44	62.1 ± 25.2	14.4 ± 5.76	4.56 ± 2.18
SimRF	$> 10^3$	$> 10^3$	$> 10^3$	110 ± 118	47.0 ± 60.4	18.2 ± 23.1	69.3 ± 21.4	23.4 ± 8.16	8.15 ± 3.82
FAVOR+	$> 10^3$	$> 10^3$	$> 10^3$	114 ± 120	48.1 ± 61.3	17.6 ± 21.9	70.0 ± 22.4	23.9 ± 9.02	8.17 ± 3.86
Coupled PRFs	$> 10^3$	$> 10^3$	$> 10^3$	114 ± 119	46.4 ± 59.8	17.5 ± 20.5	70.1 ± 22.0	23.4 ± 8.77	8.21 ± 3.70
FAVOR++	$> 10^3$	$> 10^3$	$> 10^3$	238 ± 306	95.3 ± 120	37.5 ± 52.3	107 ± 34.8	34.7 ± 11.9	13.3 ± 5.02

Table 4: Next-token CE (↓) and perplexity (↓) on the UN Parallel Corpus. The first two GMProp rows are trained on this corpus. The pretraining-mixture rows are the proposals of Table 2, scored unchanged. Full table is Table 10 (Appendix G).

	Pythia-70M			Pythia-410M			OLMo-1B		
Method	16	256	4096	16	256	4096	16	256	4096
Exact attention		3.33 (27.9)			2.50 (12.2)			2.16 (8.70)	
GMProp, with gauge	**5.10 (164)**	**4.82 (124)**	**4.75 (115)**	**4.56 (95.3)**	**4.03 (56.2)**	**3.88 (48.5)**	**4.54 (93.4)**	**4.19 (65.8)**	**4.01 (55.0)**
GMProp, no gauge	5.50 (245)	5.16 (174)	4.98 (145)	5.47 (238)	4.78 (119)	4.54 (94.2)	5.87 (355)	4.91 (136)	4.56 (95.2)
GMProp, with gauge, pretraining mixture	5.38 (216)	5.13 (170)	4.99 (146)	5.09 (162)	4.62 (101)	4.43 (84.3)	5.70 (298)	5.14 (170)	4.76 (117)
GMProp, no gauge, pretraining mixture	5.67 (290)	5.32 (205)	5.12 (167)	5.69 (295)	4.92 (138)	4.73 (114)	6.12 (453)	5.39 (219)	5.02 (152)
FAVOR#	6.62 (747)	6.06 (429)	5.62 (275)	7.69 ($> 10^3$)	6.59 (726)	5.62 (277)	8.45 ($> 10^3$)	7.20 ($> 10^3$)	6.47 (645)
SimRF	8.47 ($> 10^3$)	7.30 ($> 10^3$)	7.17 ($> 10^3$)	7.72 ($> 10^3$)	7.67 ($> 10^3$)	7.49 ($> 10^3$)	8.18 ($> 10^3$)	7.22 ($> 10^3$)	6.69 (802)
FAVOR+	8.55 ($> 10^3$)	7.34 ($> 10^3$)	7.18 ($> 10^3$)	7.75 ($> 10^3$)	7.82 ($> 10^3$)	7.43 ($> 10^3$)	8.29 ($> 10^3$)	7.25 ($> 10^3$)	6.68 (801)
Coupled PRFs	8.17 ($> 10^3$)	7.49 ($> 10^3$)	7.30 ($> 10^3$)	7.87 ($> 10^3$)	7.88 ($> 10^3$)	7.24 ($> 10^3$)	8.27 ($> 10^3$)	7.13 ($> 10^3$)	6.62 (749)
FAVOR++	10.2 ($> 10^3$)	8.63 ($> 10^3$)	8.15 ($> 10^3$)	8.80 ($> 10^3$)	8.79 ($> 10^3$)	8.23 ($> 10^3$)	8.64 ($> 10^3$)	7.36 ($> 10^3$)	6.82 (916)

4.4 Cost

At L=2048 on Pythia-70M, the GMProp, FAVOR+, SimRF and Coupled-PRF rows spend 1.72–1.79×10^{11} attention FLOPs against exact attention's 4.12×10^{11}. FAVOR++ and FAVOR# come in *above* it. Per generated token the recurring state is a constant 6.20MiB against a KV cache that grows to 48.0MiB. The full accounting, offline and online, is in Appendix H.

5 Related work

Beyond the baselines of section 4, this section compares GMProp with two related lines of work on unbiasedness for the kernel, adaptation done once offline rather than from the current queries and keys, and an $O(Lmd)$ prefill with $O(md)$ per decoded token.

Random-feature attention. FAVOR+ (Choromanski et al., 2021; Rahimi & Recht, 2007) draws its features from an isotropic Gaussian, and GMProp reduces to it when every mixture component is $\mathcal{N}(0, I_d)$ and $A = I$ (Proposition 2). FAVOR++ and FAVOR# (Likhosherstov et al., 2022; 2023) adapt the feature map inside every forward pass, so each generated token recomputes the features of the whole context (Table 13). Hybrid RFs (Choromanski et al., 2022) correlate the draws as SimRF and Coupled PRFs do. GMProp instead fits the law of each draw once offline, so each decoded token keeps the constant $O(md)$ cost that the refitting methods lose.

Learned feature maps. One variant of KL-Transformer (Pal Chowdhury et al., 2022) draws positive random features from a learned Gaussian mixture, as GMProp does, but without the importance weight, so the mixture itself defines a new kernel instead of estimating $e^{q^\top k}$, and with its gradient reaching the mixture only along the pathwise route, whereas GMProp samples the component index and trains the mixing weights through the importance weight rather than through that sampled choice. DARKFormer (Farzam et al., 2026) learns a Gaussian proposal $\mathcal{N}(0, \Sigma)$ and trains W_Q, W_K and W_V with it. Its estimate is unbiased for a different kernel, $e^{q^\top \Sigma k}$. The Coulomb particle model (Badiei Khuzani et al., 2026) learns m features, freezes them and trains the rest of the network. Nishikawa et al. (2025) learn fixed feature points and weights by distilling a pretrained model layer by layer, and DiJiang (Chen et al., 2024) learns weights on fixed quasi-Monte Carlo points and fine-tunes the model. GMProp keeps the kernel, because its importance weight fixes the mean of the estimate at $e^{q^\top k}$ and leaves the mixture only its variance to change, and it keeps the pretrained weights and uses the same proposal at every m.

Table 5: Costs are per head at sequence length L, feature count m and head dimension d, as each method's own paper states them, and decode is the cost of one token after a prompt of that length. The unbiased rows also pay a one-time $O(md^2)$ feature draw (Appendix I). A *modified kernel* is one other than $e^{q^\top k}$, a *whole softmax* target is the attention output, and a *self-normalized* estimator is a ratio of two estimates, consistent but biased at finite m. RA draws m_{RA} samples per query, and EVA uses J proposals and $|E|$ exactly evaluated positions per query.

Method	Unbiased for kernel $e^{q^\top k}$	Prefill complexity	Decode complexity (per token)
FAVOR+ (Choromanski et al., 2021)	Yes	$O(Lmd)$	$O(md)$
FAVOR++ (Likhosherstov et al., 2022)	Yes	$O(Lmd) + O(Ld)$	$O(Lmd) + O(Ld)$
FAVOR# (Likhosherstov et al., 2023)	Yes	$O(Lmd)$ $+ O(Ld^2 + md^2 + d^3)$	$O\big(L(md + d^2)$ $+ md^2 + d^3\big)$
SimRF (Reid et al., 2023)	Yes	$O(Lmd)$	$O(md)$
Coupled PRFs (Reid et al., 2025)	Yes	$O(Lmd)$	$O(md)$
DARKFormer (Farzam et al., 2026)	No, modified kernel	$O(Lmd)$	$O(md)$
Coulomb particles (Badiei Khuzani et al., 2026)	No	$O(Lmd)$, m fixed at training	$O(md)$
Randomized attention (RA) (Zheng et al., 2022)	No, whole softmax	$O(m_{\mathrm{RA}} L^2 d)$	$O(m_{\mathrm{RA}} Ld)$
LARA and EVA, causal variant EVA (Zheng et al., 2022; 2023)	No, self-normalized	$O(LJd)$	$O(Jmd + \|E\|d)$
GMProp (ours)	Yes	$O(Lmd)$	$O(md)$

6 Conclusion

GMProp matches a pretrained model's query and key anisotropy by learning only the *sampling law* of a PRF estimator, as a small Gaussian mixture, and leaves the network exact. It is unbiased for the attention kernel, leaves the pretrained model weights bit-identical, prefills in $O(Lmd)$ and decodes each token in $O(md)$, and surpasses every admitted baseline across each model's own pretraining mixture and three scales. The same proposals, in a cross-corpus test with nothing refit, stay ahead of FAVOR#, since the pair sums they cover depend mainly on the model's projections, not on the corpus. Over the tested range the residual gap to exact attention decays slowly with m: $256\times$ more features close 0.460 of 2.27 nats on Pythia-70M (Table 2).

AI USE STATEMENT

Large language models were used as general-purpose assistants for editing prose and for routine coding support in the experimental harness. All claims, proofs and measurements were formulated and verified by the authors, who take full responsibility for the content of this paper.

ETHICS STATEMENT

This work studies an inference-time approximation to attention in existing pretrained language models. All models (`pythia-70m`, `pythia-410m`, `OLMo-1B-hf`) and corpora (the copyright-stripped Pile, Dolma v1.5, the UN Parallel Corpus v1.0) are publicly released for research use, and no new data was collected and no human subjects were involved. Because the pretrained weights are never modified and the estimator is unbiased for the model's own attention kernel, the method does not introduce a new capability or a new alignment surface. It inherits whatever biases and failure modes the underlying pretrained models already carry. The efficiency gains it offers apply equally to beneficial and harmful deployments of those models, which is the standard dual-use position of work on efficient inference.

REPRODUCIBILITY STATEMENT

Every claim in section 4 is reproducible from the description in Appendix G, which gives the training protocol, hyperparameters, set construction and sizes, feature counts and seeds. Proofs of all stated results are in Appendix A, with the assumptions stated in the theorem statements themselves. The models and corpora are public. Cost measurements (Appendix H) record the hardware, precision, batch shape and sequence length they were taken at, since those choices — not only the mechanism — determine the wall-clock column. Source code will be released with the camera-ready version.

REFERENCES

Ömer Deniz Akyildiz and Joaquín Míguez. Convergence rates for optimised adaptive importance samplers. *Statistics and Computing*, 31(12), 2021.

Masoud Badiei Khuzani, Sharath Honnaiah, Atiq Islam, Alex Cozzi, and Abraham Bagherjeiran. A coulomb particle model for learning kernel attention in transformers. *arXiv preprint arXiv:2607.23869*, 2026. Workshop on High-dimensional Learning Dynamics (HiLD), ICML.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Pierre Blanchard, Desmond J. Higham, and Nicholas J. Higham. Accurately computing the log-sum-exp and softmax functions. *IMA Journal of Numerical Analysis*, 41(4):2311–2330, 2021.

Olivier Cappé, Randal Douc, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459, 2008.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Hanting Chen, Zhicheng Liu, Xutao Wang, Yuchuan Tian, and Yunhe Wang. DiJiang: Efficient large language models through compact kernelization. In *International Conference on Machine Learning (ICML)*, 2024.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations (ICLR)*, 2021.

Krzysztof Choromanski, Haoxian Chen, Han Lin, Yuanzhe Ma, Arijit Sehanobish, Deepali Jain, Michael S. Ryoo, Jake Varley, Andy Zeng, Valerii Likhosherstov, Dmitry Kalashnikov, Vikas Sindhwani, and Adrian Weller. Hybrid random features. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2110.04367.

Bernard Delyon and François Portier. Asymptotic optimality of adaptive importance sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Bernard Delyon and François Portier. Safe adaptive importance sampling: A mixture approach. *The Annals of Statistics*, 49(2):885–917, 2021.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, 2019.

Amirhossein Farzam, Hossein Mobahi, Nolan Andrew Miller, and Luke Sernau. Data-aware random feature kernel for transformers. *arXiv preprint arXiv:2603.04127*, 2026.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Adam Goliński, Frank Wood, and Tom Rainforth. Amortized Monte Carlo integration. In *International Conference on Machine Learning (ICML)*, pp. 2309–2318, 2019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Hera Y. He and Art B. Owen. Optimal mixture weights in multiple importance sampling. *arXiv preprint arXiv:1411.3954*, 2014.

Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

Reiichiro Kawai. Optimizing adaptive importance sampling by stochastic approximation. *SIAM Journal on Scientific Computing*, 40(4):A2774–A2800, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Anna Korba and François Portier. Adaptive importance sampling meets mirror descent: A bias-variance tradeoff. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 11503–11527, 2022.

Vincent Lemaire and Gilles Pagès. Unconstrained recursive importance sampling. *The Annals of Applied Probability*, 20(3):1029–1067, 2010.

Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random Fourier features. *Journal of Machine Learning Research*, 22, 2021.

Valerii Likhosherstov, Krzysztof Choromanski, Avinava Dubey, Frederick Liu, Tamas Sarlos, and Adrian Weller. Chefs' random tables: Non-trigonometric random features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2205.15317. FAVOR++ / OPRF.

Valerii Likhosherstov, Krzysztof Choromanski, Avinava Dubey, Frederick Liu, Tamas Sarlos, and Adrian Weller. FAVOR#: Sharp attention kernel approximations via new classes of positive random features. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2302.00787.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics*, 38(5), 2019.

Naoki Nishikawa, Rei Higuchi, and Taiji Suzuki. Degrees of freedom for linear attention: Distilling softmax attention with optimal feature efficiency. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Peder A. Olsen and John R. Hershey. Bhattacharyya error and divergence using variational importance sampling. In *Interspeech*, pp. 46–49, 2007.

Art B. Owen. *Monte Carlo Theory, Methods and Examples*. Stanford University, 2013. https://artowen.su.domains/mc/.

Sankalan Pal Chowdhury, Adamos Solomou, Avinava Dubey, and Mrinmaya Sachan. Learning the transformer kernel. *Transactions on Machine Learning Research (TMLR)*, 2022.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations (ICLR)*, 2021.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.

Isaac Reid, Krzysztof Marcin Choromanski, Valerii Likhosherstov, and Adrian Weller. Simplex random features. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2301.13856.

Isaac Reid, Stratis Markou, Krzysztof Choromanski, Richard E. Turner, and Adrian Weller. Variance-reducing couplings for random features. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2405.16541. PNC/PM couplings.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, pp. 1278–1286, 2014.

Ernest K. Ryu and Stephen P. Boyd. Adaptive importance sampling via stochastic convex programming. Technical report, Stanford University, 2014. https://web.stanford.edu/~boyd/papers/adaMC.html.

Luke Sernau, Silvano Bonacina, and Rif A. Saurous. All random features representations are equivalent. *arXiv preprint arXiv:2406.18802*, 2024.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: an open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pp. 2214–2218, 2012.

William Timkey and Marten van Schijndel. All bark and no bite: Rogue dimensions in transformer language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4527–4546, 2021.

Tim van Erven and Peter Harremoës. Rényi divergence and kullback–leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25, 2024.

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

Ming Xu, Matias Quiroz, Robert Kohn, and Scott A. Sisson. Variance reduction properties of the reparameterization trick. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Felix X. X. Yu, Ananda Theertha Suresh, Krzysztof Choromanski, Daniel Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Michael Zhang, Simran Arora, Rahul Chalamala, Alan Wu, Benjamin Spector, Aaryan Singhal, Krithik Ramesh, and Christopher Ré. LoLCATs: On low-rank linearizing of large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Lin Zheng, Chong Wang, and Lingpeng Kong. Linear complexity randomized self-attention mechanism. In *International Conference on Machine Learning (ICML)*, 2022.

Lin Zheng, Jianbo Yuan, Chong Wang, and Lingpeng Kong. Efficient attention via control variates. In *International Conference on Learning Representations (ICLR)*, 2023. URL `https://arxiv.org/abs/2302.04542`.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The United Nations Parallel Corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pp. 3530–3534, 2016.

APPENDIX CONTENTS

A Proofs 14
A.1 The variance-optimal proposal 14
A.2 Guarantees by construction 15
A.3 Capacity 15
B The derivatives of $\log r_\theta$ at fixed ω 20
C The deployed feature map 20
D Direct variance minimization: a negative result 21
E The log-kernel MSE 26
F Corpora 27
G Experimental details 28
H Cost accounting 35
H.1 The cost of the gauge 35
H.2 Cost of learning 35
H.3 Cost of inference: prefill 36
H.4 Cost of inference: decode 36

H.5 The accuracy-aware cost frontier 37
I Extended related work 37

A PROOFS

A.1 THE VARIANCE-OPTIMAL PROPOSAL

Theorem 1. Fix pair weights $\mathcal{P}_{ij}$ as in the theorem and a density $\psi > 0$ on $\mathbb{R}^d$. Take the second moment of a single draw of section 3.4 with ψ in place of r_θ, so that the term is $w(\omega)\phi_\omega^+(q)\phi_\omega^+(k)$ with $w = p_I/\psi$. For one pair, $\mathbb{E}_{\omega\sim\psi}[w(\omega)^2\phi_\omega^+(q)^2\phi_\omega^+(k)^2] = \int p_I(\omega)^2\phi_\omega^+(q)^2\phi_\omega^+(k)^2/\psi(\omega)\,d\omega$, and averaging over the pairs, with the $1/m$ of the m-draw estimator, defines

$$S(\psi) = \frac{1}{m}\int \frac{p_I^2\,Q}{\psi}, \qquad Q(\omega) := \mathbb{E}_{q,k}\big[\phi_\omega^+(q)^2\,\phi_\omega^+(k)^2\big] = \sum_{i,j}\mathcal{P}_{ij}\,\phi_\omega^+(q_i)^2\,\phi_\omega^+(k_j)^2, \tag{19}$$

the joint second moment of the squared features. Nothing is assumed about the dependence between q and k. Put $f(\omega) := p_I(\omega)\sqrt{Q(\omega)}$. By Cauchy–Schwarz, for every density ψ,

$$\Big(\int f\Big)^2 = \Big(\int \frac{f}{\sqrt{\psi}}\sqrt{\psi}\Big)^2 \le \Big(\int \frac{f^2}{\psi}\Big)\Big(\int \psi\Big) = m\,S(\psi),$$

with equality iff $f/\sqrt{\psi} \propto \sqrt{\psi}$ almost everywhere, i.e. $\psi \propto f$. Hence S is minimized over all densities, uniquely up to a null set, by

$$\psi^*(\omega) \propto p_I(\omega)\sqrt{Q(\omega)} = \sqrt{p_I(\omega)^2 Q(\omega)}, \tag{20}$$

with minimum $S(\psi^*) = \frac{1}{m}(\int f)^2$. The expected Monte Carlo variance of one term, the objective of equation 11, is $m\,S(\psi) - \mathbb{E}_{q,k}\kappa(q,k)^2$ by unbiasedness (Proposition 1), and the subtracted term does not depend on ψ, so ψ^* minimizes it as well.

It remains to compute $p_I^2 Q$. Completing the square in the exponent of a single pair, with $\|q+k\|^2 - \|q\|^2 - \|k\|^2 = 2q^\top k$,

$$\begin{aligned} p_I(\omega)^2\,\phi_\omega^+(q)^2\,\phi_\omega^+(k)^2 &= (2\pi)^{-d}\,e^{-\|\omega\|^2 + 2\omega^\top(q+k) - \|q\|^2 - \|k\|^2} = (2\pi)^{-d}\,\kappa(q,k)^2\,e^{-\|\omega-(q+k)\|^2} \\ &= \kappa(q,k)^2\,\mathcal{N}(\omega \mid q+k, I_d)^2 = (4\pi)^{-d/2}\,\kappa(q,k)^2\,\mathcal{N}(\omega \mid q+k, \tfrac{1}{2}I_d), \end{aligned}$$

the square of equation 8, using in the last step $\mathcal{N}(\omega \mid x, I_d)^2 = (4\pi)^{-d/2}\mathcal{N}(\omega \mid x, \frac{1}{2}I_d)$, which follows from $(2\pi)^{-d} = (4\pi)^{-d/2}\pi^{-d/2}$. Averaging over the pairs with the weights $\mathcal{P}_{ij}$,

$$p_I^2\,Q = \sum_{i,j}\mathcal{P}_{ij}\,\kappa(q_i,k_j)^2\,\mathcal{N}(\cdot \mid q_i+k_j, I_d)^2 = (4\pi)^{-d/2}\,h, \tag{21}$$

with h the Gaussian mixture under the root in equation 18, one component at each pair sum $q_i + k_j$ with covariance $\frac{1}{2}I_d$ and weight $\mathcal{P}_{ij}\,\kappa(q_i,k_j)^2$. Subadditivity of the root gives

$$\sqrt{h} \le (4\pi)^{d/4}\sum_{i,j}\sqrt{\mathcal{P}_{ij}}\,\kappa(q_i,k_j)\,\mathcal{N}(\cdot \mid q_i+k_j, I_d),$$

so $0 < \int\sqrt{h} < \infty$ and ψ^* is a density. Taking the root in equation 20, $\psi^* \propto \sqrt{p_I^2 Q} = (4\pi)^{-d/4}\sqrt{h}$, and the constant cancels on normalizing by $\int\sqrt{h}$, which gives equation 18. □

Notation for the proofs. Enumerate the distinct pair sums $q_i + k_j$ of the pairs with $\mathcal{P}_{ij} > 0$ as $x_1, \dots, x_P$, and merge the weights of the pairs sharing a sum, $\bar\nu_p := \sum_{(i,j):\,q_i+k_j=x_p}\mathcal{P}_{ij}\,\kappa(q_i,k_j)^2 > 0$, so that $\sum_p \bar\nu_p = \mathbb{E}_{q,k}\kappa(q,k)^2$. The mixture h under the root in equation 18 is then, as a function on $\mathbb{R}^d$,

$$h = \sum_{p=1}^{P}\bar\nu_p\,\mathcal{N}(\cdot \mid x_p, \tfrac{1}{2}I_d), \qquad Z := \int\sqrt{h}, \qquad \psi^* = \frac{\sqrt{h}}{Z}, \tag{22}$$

with one component per distinct centre, and by equation 21 $f(\omega) = p_I(\omega)\sqrt{Q(\omega)} = (4\pi)^{-d/4}\sqrt{h(\omega)}$, so that $S(\psi) = \frac{1}{m}\int f^2/\psi$ in equation 19.

A.2 GUARANTEES BY CONSTRUCTION

The first guarantee holds for every parameter value and every gauge, so nothing about it depends on training.

Proposition 1 (Unbiasedness)**.** *For every θ and every invertible gauge A, r_θ is a normalized density with full support on $\mathbb{R}^d$, and equation 5 satisfies $\mathbb{E}_{\omega_{1:m}\sim r_\theta}[\hat{\kappa}_\theta(q,k)] = \kappa(q,k)$ and $\hat{\kappa}_\theta > 0$.*

Proof of Proposition 1. Fix any θ and any invertible gauge A. The density claim is immediate: equation 6 is a convex combination of Gaussian densities with strictly positive mixing coefficients $\lambda = \mathrm{softmax}(\nu)$ and strictly positive widths $\sigma_c = e^{\log \sigma_c}$, hence a strictly positive density integrating to 1, for every value of the means.

Consequently r_θ, like p_I, is positive on all of $\mathbb{R}^d$, so the weight p_I/r_θ is finite everywhere and, for $\omega \sim r_\theta$,

$$\mathbb{E}_{\omega\sim r_\theta}\Big[\tfrac{p_I(\omega)}{r_\theta(\omega)}\phi^+_\omega(q)\phi^+_\omega(k)\Big] = \int \phi^+_\omega(q)\phi^+_\omega(k)\, p_I(\omega)\, d\omega = e^{q^\top k},$$

so equation 5 is unbiased, and it is a nonnegative average of positive terms, hence positive. By equation 9 the gauge leaves $q^\top k$ pointwise unchanged, so the estimand $e^{q^\top k}$ is identical in gauged and ungauged coordinates. None of this uses convergence of training, Gaussianity of the data, or the value of m. □

Two further guarantees follow from the same construction.

Proposition 2 (Exact FAVOR+ initialization)**.** *At the neutral initialization, $\mu = 0$, $\sigma = 1$ and $A = I$, every component of equation 6 equals $\mathcal{N}(0, I_d)$, so $r_\theta = p_I$ for any λ and $\log w \equiv 0$ to floating point.*

Proof. At the neutral initialization μ_c=0 and σ_c=1 make every component of equation 6 equal to $\mathcal{N}(0, I_d) = p_I$, so any convex combination of them is p_I regardless of λ. With A=I the gauge is the identity map. Hence $\log w = \log p_I - \log r_\theta \equiv 0$ and equation 5 is the FAVOR+ estimator under the same marginal proposal. Whether the m draws are i.i.d. or form an orthogonal block is a choice of the sampler that θ does not touch. □

Proposition 3 (Feature-count invariance)**.** *Unbiasedness holds for every $m \geq 1$, and the pair-averaged second moment is $S(r_\theta) = \frac{1}{m}\int f^2/r_\theta$, in which m enters only as the prefactor $1/m$. Hence the r_θ minimizing S is the same for every m, and a single checkpoint may be evaluated at any feature count.*

Proof of Proposition 3. Proposition 1 holds for each summand of equation 5 individually, so the mean of any $m \geq 1$ draws is unbiased. The pair-averaged second moment of equation 19, with f as in the notation above, is $S(r_\theta) = \frac{1}{m}\int f^2/r_\theta$. The integral is independent of m, which appears only as $1/m$. Hence the r_θ minimizing S is the same for every m. This is a statement about S: the surrogates actually trained in section 3.4, the log-kernel MSE and the next-token cross-entropy, are finite-m functionals of the estimator equation 5, so their minimizers need not be independent of m, and we do not claim a trained proposal is optimal at a budget other than the one it was fit at. What does carry to every m is the proposal: r_θ contains no m and is unbiased at each m, which is all that scoring one proposal at several feature counts requires. □

A.3 CAPACITY

Write

$$V(\psi) := \mathbb{E}_{q,k}\big[\mathrm{Var}_{\omega\sim\psi}\big(w(\omega)\,\phi^+_\omega(q)\,\phi^+_\omega(k)\big)\big], \qquad V_C := \inf_\theta V(r_\theta), \tag{23}$$

for the expected Monte Carlo variance of a proposal ψ, the functional Theorem 1 minimizes, and for the smallest value a mixture equation 6 with C components reaches. Capacity is how V_C approaches $V(\psi^*)$ as C grows.

Theorem 2 (Capacity)**.** *Let the pair weights $\mathcal{P}_{ij}$ be as in Theorem 1. Then*

$$V_C \;\geq\; V(\psi^*) \text{ for every } C, \qquad V_{C+1} \;\leq\; V_C, \qquad \lim_{C\to\infty} V_C \;=\; V(\psi^*). \tag{24}$$

If the query set and the key set are the same n points and the weights are uniform over all n^2 pairs, then $V_C = V(\psi^*)$ for every $C \geq n$. Moreover, for every partition of the pair sums $q_i + k_j$ into C groups with centres $c_1, \ldots, c_C$, radii at most δ and weights $W_1, \ldots, W_C$, W_c the total weight $\mathcal{P}_{ij}\kappa(q_i, k_j)^2$ of the pairs whose sum lies in group c,

$$V_C \;\leq\; V(\psi^*) + (B_C - 1)\big(V(\psi^*) + \mathbb{E}_{q,k}\kappa(q,k)^2\big),$$
$$B_C = \inf_{s>1}\; s^d\, e^{\delta^2/(2(s^2-1))}\, \frac{(4\pi)^{d/4}\sum_{c\leq C}\sqrt{W_c}}{\int \sqrt{h}}, \tag{25}$$

with h the Gaussian mixture of equation 22.

The rate equation 25 is a covering bound: exact when all pair sums coincide, tight for well-separated tight groups, and vacuous once $\delta\sqrt{d}$ is large, so it certifies the approach to $V(\psi^*)$ for tightly clustered pair sums rather than explaining the $C = 16$ of section 4 in $d = 64$.

This section proves Theorem 2 with the notation of Appendix A.1, through a divergence identity (Lemma 1), the symmetric-case clause (Proposition 4) and the covering rate (Proposition 5).

Lemma 1 (The second-moment ratio is a Rényi-2 divergence)**.** *For every density $\psi > 0$ on $\mathbb{R}^d$,*

$$\frac{S(\psi)}{S(\psi^*)} \;=\; \int \frac{(\psi^*)^2}{\psi} \;=\; 1 + \chi^2(\psi^*\|\psi) \;=\; e^{D_2(\psi^*\|\psi)} \;\geq\; 1,$$

with equality if and only if $\psi = \psi^*$ almost everywhere.

Minimizing the second moment is thus minimizing a χ^2 divergence (van Erven & Harremoës, 2014; Owen, 2013), a multiplicative criterion under which r_θ must be nowhere much smaller than ψ^*.

Proposition 4 (Exact optimality in the symmetric case)**.** *If the queries and the keys are the same n points $x_1, \ldots, x_n$ and the weights are uniform over all n^2 pairs, then $\psi^* = \sum_{i=1}^n \pi_i \mathcal{N}(\omega \mid 2x_i, I_d)$ with $\pi_i \propto e^{\|x_i\|^2}$. That is a GMProp proposal with means $2x_i$, unit widths and mixing coefficients π, so $\psi^* = r_\theta$ for some $\theta \in \Theta(C)$ whenever $C \geq n$, and $\rho(C) = 1$ exactly.*

Standing assumptions and notation. The pair weights $\mathcal{P}_{ij}$ are as in Theorem 1, and $x_1, \ldots, x_P$, $\bar{\nu}_p$, h and Z are as in equation 22. Each summand of h is a strictly positive C^∞ function, so $h > 0$. Then $Z > 0$ since $h > 0$, and $Z \leq (4\pi)^{d/4}\sum_p \sqrt{\bar{\nu}_p} < \infty$ by Lemma 3 below, so

$$\psi^* = \frac{\sqrt{h}}{Z} \tag{26}$$

is a strictly positive C^∞ probability density. $\Theta(C)$ denotes the C-component GMProp parameters of equation 6 with gauge $A = I$, and $\mathcal{F}(C) := \{r_\theta : \theta \in \Theta(C)\}$ is the corresponding set of densities: all finite mixtures $\sum_{c\leq C}\lambda_c \mathcal{N}(\cdot \mid \mu_c, \sigma_c^2 I_d)$ with λ strictly positive and summing to one. The attainable ratio is

$$\rho(C) \;:=\; \inf_{\theta\in\Theta(C)}\; \frac{S(r_\theta)}{S(\psi^*)}, \tag{27}$$

and what the proof below establishes is that $\rho(C) \geq 1$ for all C, that ρ is non-increasing in C, and that for every $\varepsilon \in (0, 1]$ there is a finite $C(\varepsilon)$ with $\rho(C) \leq 1 + \varepsilon$ whenever $C \geq C(\varepsilon)$. The symmetric-case clause of Theorem 2 is Proposition 4 and the rate equation 25 is Proposition 5. The V of equation 23 is $V(\psi) = m\,S(\psi) - \mathbb{E}_{q,k}\kappa(q,k)^2$, the expected Monte Carlo variance of one term, and $V_C = \inf_{\theta\in\Theta(C)} V(r_\theta)$. The subtracted term is the same for every ψ, so $V_C - V(\psi^*) = m\,(\rho(C) - 1)\,S(\psi^*)$ with $m\,S(\psi^*) = V(\psi^*) + \mathbb{E}_{q,k}\kappa(q,k)^2$, and these three properties of ρ are the three claims of equation 24: the excess variance is nonnegative, non-increasing in C, and tends to zero; $\rho(C) \leq B_C$ is equation 25. Fixing $A = I$ is without loss, but not because the gauge is inert on the optimum: equation 9 leaves κ pointwise unchanged while mapping the pair sums to $Aq_i + A^{-\top}k_j$, so ψ^* changes, which is why a gauge can reduce variance at all. What makes $A = I$ harmless is that every statement below is proved for an *arbitrary* choice of the pair weights $\mathcal{P}_{ij}$, and a gauged head is another such choice: the bounds apply verbatim in the gauged frame, against that frame's own ψ^*. We use that $\mathrm{softmax}$ maps onto the *open* simplex, so any strictly positive weight vector is realizable exactly, and that $\sigma = e^{\log\sigma}$ realizes any strictly positive width.

Proof of Lemma 1. $Z_f := \int f = (4\pi)^{-d/4} Z \in (0, \infty)$ by equation 22, so $\psi^* = f/Z_f$ agrees with equation 26 and is a density. Then

$$S(\psi^*) = \frac{1}{m} \int \frac{f^2}{f/Z_f} = \frac{Z_f}{m} \int f = \frac{Z_f^2}{m}, \qquad S(\psi) = \frac{1}{m} \int \frac{f^2}{\psi} = \frac{Z_f^2}{m} \int \frac{(f/Z_f)^2}{\psi},$$

whence $S(\psi)/S(\psi^*) = \int (\psi^*)^2/\psi$, where both sides may be $+\infty$. Expanding the nonnegative integrand $(\psi^* - \psi)^2/\psi$ and using $\int \psi^* = \int \psi = 1$ gives $\chi^2(\psi^* \| \psi) = \int (\psi^*)^2/\psi - 1$, and $D_2(\psi^* \| \psi) = \log \int (\psi^*)^2/\psi$ is the order-2 Rényi divergence (van Erven & Harremoës, 2014). Finally, by Cauchy–Schwarz,

$$1 = \Big(\int \psi^* \Big)^2 = \Big(\int \frac{\psi^*}{\sqrt{\psi}} \sqrt{\psi} \Big)^2 \le \Big(\int \frac{(\psi^*)^2}{\psi} \Big) \Big(\int \psi \Big) = \int \frac{(\psi^*)^2}{\psi},$$

with equality iff $\psi^*/\sqrt{\psi} = c\sqrt{\psi}$ a.e. for a constant c, i.e. $\psi^* = c\psi$ a.e. Integrating forces $c = 1$. □

Lemma 2 (Nesting)**.** $\mathcal{F}(C) \subseteq \mathcal{F}(C+1)$. *Consequently ρ of equation 27 is non-increasing in C.*

Proof. Given θ with components $1, \dots, C$, duplicate component C and replace λ_C by two mixing coefficients $\lambda_C/2$. All mixing coefficients stay strictly positive and sum to one, and the density is unchanged. An infimum over a larger set is no larger, giving the monotonicity. □

Lemma 3 (A dominating defensive mixture in the family)**.** *Let*

$$G := \sum_{p=1}^{P} \sqrt{\bar{\nu}_p}\, \mathcal{N}(\cdot \mid x_p, I_d), \qquad Z_G := \sum_{p=1}^{P} \sqrt{\bar{\nu}_p},$$

and $g := G/Z_G$. Then $Z_G = \int G \in (0, \infty)$, $g \in \mathcal{F}(P)$, $\sqrt{h} \le (4\pi)^{d/4} G$ pointwise, and $g \ge \kappa_G^{-1} \psi^*$ pointwise with $\kappa_G := (4\pi)^{d/4} Z_G / Z \in [1, \infty)$.

Proof. Subadditivity of the square root on nonnegative summands, $\sqrt{\sum_\nu x_\nu} \le \sum_\nu \sqrt{x_\nu}$, which holds because squaring both sides leaves only nonnegative cross terms, gives $\sqrt{h} \le \sum_p \sqrt{\bar{\nu}_p} \sqrt{\mathcal{N}(\cdot \mid x_p, \frac{1}{2} I_d)}$, and $\sqrt{\mathcal{N}(\omega \mid x, \frac{1}{2} I_d)} = \pi^{-d/4} e^{-\|\omega - x\|^2/2} = (4\pi)^{d/4} \mathcal{N}(\omega \mid x, I_d)$, so the right-hand side is $(4\pi)^{d/4} G$. Each component of G integrates to one, so $\int G = Z_G$, a finite positive sum. Therefore g is a density, and it is a mixture of P Gaussians with covariance I_d and strictly positive mixing coefficients, that is, a GMProp proposal with means x_p and unit widths, so $g \in \mathcal{F}(P)$. Finally $\psi^* = \sqrt{h}/Z \le (4\pi)^{d/4} G/Z = \kappa_G\, g$, and integrating $\sqrt{h} \le (4\pi)^{d/4} G$ gives $Z \le (4\pi)^{d/4} Z_G$, i.e. $\kappa_G \ge 1$. □

Lemma 4 (Score bound and local oscillation)**.** *Let $u = \sum_\ell w_\ell \mathcal{N}(\cdot \mid m_\ell, \sigma^2 I_d)$ with $w_\ell > 0$ and $\sigma > 0$. Then*

$$\nabla \log u(\omega) = \sigma^{-2} \sum_\ell \rho_\ell(\omega)(m_\ell - \omega), \qquad \rho_\ell(\omega) = \frac{w_\ell \mathcal{N}(\omega \mid m_\ell, \sigma^2 I_d)}{u(\omega)},$$

where $\rho(\omega)$ is a probability vector, so $\|\nabla \log u(\omega)\| \le \sigma^{-2} \max_\ell \|m_\ell - \omega\|$. Consequently, with $\rho_{\max} := \max_p \|x_p\|$, $B_R := [-R, R]^d$ and $L_R := \rho_{\max} + R\sqrt{d}$,

$$\sup_{\omega \in B_R} \|\nabla \log \psi^*(\omega)\| \le L_R, \qquad \operatorname*{osc}_S \log \psi^* \le L_R\, h \text{ for every convex } S \subseteq B_R \text{ of diameter} \le h.$$

Proof. Differentiating u termwise, $\nabla u(\omega) = \sigma^{-2} \sum_\ell w_\ell \mathcal{N}(\omega \mid m_\ell, \sigma^2 I_d)(m_\ell - \omega)$. Dividing by $u(\omega) > 0$ gives the stated score, and ρ is a probability vector by construction, so the score is σ^{-2} times a convex combination of the vectors $m_\ell - \omega$ and its norm is at most σ^{-2} times their maximum norm. By equation 26, $\log \psi^* = \frac{1}{2} \log h - \log Z$ with h a mixture of covariance $\frac{1}{2} I_d$ at the x_p, so $\|\nabla \log \psi^*(\omega)\| \le \frac{1}{2} \cdot 2 \max_p \|x_p - \omega\| \le \rho_{\max} + \|\omega\|$, and $\|\omega\| \le R\sqrt{d}$ on B_R. The oscillation bound is the mean value inequality applied along the segment joining any two points of the convex set S. □

Proof of Theorem 2. The lower bound $\rho(C) \geq 1$ is Lemma 1, monotonicity in C is Lemma 2, the symmetric clause is Proposition 4, and the rate equation 25 is Proposition 5 with $V_C - V(\psi^*) = m(\rho(C) - 1)S(\psi^*)$ and $mS(\psi^*) = V(\psi^*) + \mathbb{E}_{q,k}\kappa^2$. We prove the limit, $\rho(C) \leq 1 + \varepsilon$ for all $C \geq C(\varepsilon)$, by an explicit construction: a defensive part that dominates the tails, and a kernel density estimate that resolves a large box. Fix $\varepsilon \in (0, 1]$ and set $\delta := \varepsilon/4$.

Choice of R, then s, then h. Since ψ^* is a probability density, $\psi^*(\mathbb{R}^d \setminus B_R) \to 0$ as $R \to \infty$. With κ_G from Lemma 3, which does not depend on R, fix $R \geq 1$ with $\psi^*(\mathbb{R}^d \setminus B_R) \leq \varepsilon^2/(16\kappa_G)$. Put $R^+ := R + 1$ and let $L^+ := L_{R^+} = \rho_{\max} + R^+\sqrt{d}$ be the score bound of Lemma 4 on the enlarged box B_{R^+}. The enlargement is what keeps a unit-radius smoothing window around any $\omega \in B_R$ inside the region the components cover. Next fix a bandwidth $s \in (0, 1]$ with

$$\gamma_s := \int_{\|v\| \leq 1} e^{-L^+\|v\|}\, \mathcal{N}(v \mid 0, s^2 I_d)\, dv \;\geq\; e^{-\varepsilon/8}. \tag{28}$$

This is possible because $\gamma_s \geq e^{-L^+ s\sqrt{d}} - ds^2 \to 1$ as $s \downarrow 0$: writing $v \sim \mathcal{N}(0, s^2 I_d)$, Jensen's inequality applied to the convex map $x \mapsto e^{-L^+ x}$ with $\mathbb{E}_v[\|v\|] \leq (\mathbb{E}_v[\|v\|^2])^{1/2} = s\sqrt{d}$ gives $\mathbb{E}_v[e^{-L^+\|v\|}] \geq e^{-L^+ s\sqrt{d}}$, while the discarded part is at most $\Pr_v(\|v\| > 1) \leq \mathbb{E}_v[\|v\|^2] = ds^2$ by Markov's inequality. Finally fix $h \in (0, 1]$ with $2R^+\sqrt{d}\, h/s^2 \leq \varepsilon/8$.

The cells. Let $M := \lceil 2R^+\sqrt{d}/h \rceil$ and cut B_{R^+} into M^d congruent boxes $\mathcal{R}_b$ of side $2R^+/M \leq h/\sqrt{d}$, hence of diameter at most h. These tile B_{R^+}. Write $\mathcal{B}$ for this set of cells and μ_b for the center of $\mathcal{R}_b$. Since $\psi^* > 0$ everywhere, $\psi^*(\mathcal{R}_b) > 0$ for every b.

The proposal. Take $C := M^d + P$ components:

- for each $b \in \mathcal{B}$, a component $\mathcal{N}(\cdot \mid \mu_b, s^2 I_d)$ with weight $\lambda_b = (1 - \delta)\, \psi^*(\mathcal{R}_b)/P^+$, where $P^+ := \psi^*(B_{R^+}) \in (0, 1]$;
- the P components of g from Lemma 3, with their g-weights scaled by δ.

All weights are strictly positive and sum to $(1 - \delta) + \delta = 1$, and every component has strictly positive widths, so this is a C-component GMProp density: some $\theta \in \Theta(C)$, and we write $r := r_\theta$. By construction

$$r \;\geq\; (1 - \delta) \sum_{b \in \mathcal{B}} \frac{\psi^*(\mathcal{R}_b)}{P^+}\, \mathcal{N}(\cdot \mid \mu_b, s^2 I_d) \quad \text{and} \quad r \;\geq\; \delta\, g \qquad \text{pointwise.} \tag{29}$$

Inside B_R, step 1: the cell sum is a Riemann sum. Fix $\omega \in B_R$, let $b \in \mathcal{B}$ and $u \in \mathcal{R}_b$. Both u and μ_b lie in B_{R^+}, so $\|\omega - u\|$ and $\|\omega - \mu_b\|$ are at most $2R^+\sqrt{d}$, while $\|u - \mu_b\| \leq h$. Using $\|\omega - \mu_b\|^2 - \|\omega - u\|^2 = \langle u - \mu_b,\, 2\omega - u - \mu_b \rangle$,

$$\Big| \log \frac{\mathcal{N}(\omega \mid u, s^2 I_d)}{\mathcal{N}(\omega \mid \mu_b, s^2 I_d)} \Big| = \frac{\big|\langle u - \mu_b,\, 2\omega - u - \mu_b \rangle\big|}{2s^2} \;\leq\; \frac{h \cdot 4R^+\sqrt{d}}{2s^2} \;=\; \frac{2R^+\sqrt{d}\, h}{s^2}.$$

Hence $\mathcal{N}(\omega \mid \mu_b, s^2 I_d) \geq e^{-2R^+\sqrt{d}\, h/s^2}\, \mathcal{N}(\omega \mid u, s^2 I_d)$ for every $u \in \mathcal{R}_b$. Multiplying by $\psi^*(u)$, integrating over $\mathcal{R}_b$, and summing over the cells b, which tile B_{R^+},

$$\sum_{b \in \mathcal{B}} \frac{\psi^*(\mathcal{R}_b)}{P^+} \mathcal{N}(\omega \mid \mu_b, s^2 I_d) \;\geq\; \frac{e^{-2R^+\sqrt{d}\, h/s^2}}{P^+} \int_{B_{R^+}} \psi^*(u)\, \mathcal{N}(\omega \mid u, s^2 I_d)\, du. \tag{30}$$

Inside B_R, step 2: smoothing loses little. Still with $\omega \in B_R$, substitute $u = \omega + v$ and use $\mathcal{N}(\omega \mid \omega + v, s^2 I_d) = \mathcal{N}(v \mid 0, s^2 I_d)$. If $\|v\| \leq 1$ then $\|\omega + v\|_\infty \leq R + 1 = R^+$, so $\omega + v \in B_{R^+}$ and the segment from ω to $\omega + v$ stays in the convex set B_{R^+}. Lemma 4 then gives $\psi^*(\omega + v) \geq \psi^*(\omega) e^{-L^+\|v\|}$. Discarding the rest of the integrand, which is nonnegative,

$$\int_{B_{R^+}} \psi^*(u)\, \mathcal{N}(\omega \mid u, s^2 I_d)\, du \;\geq\; \psi^*(\omega) \int_{\|v\| \leq 1} e^{-L^+\|v\|} \mathcal{N}(v \mid 0, s^2 I_d)\, dv \;=\; \gamma_s\, \psi^*(\omega). \tag{31}$$

Combining equation 29, equation 30 and equation 31 gives $r(\omega) \geq (1-\delta)\,\gamma_s\, e^{-2R^+\sqrt{d}\,h/s^2}\,\psi^*(\omega)/P^+$, so

$$\frac{\psi^*(\omega)^2}{r(\omega)} \;\leq\; \frac{P^+ e^{2R^+\sqrt{d}\,h/s^2}}{(1-\delta)\,\gamma_s}\,\psi^*(\omega),$$

and integrating over B_R with $P^+ \leq 1$ and $\psi^*(B_R) \leq 1$,

$$\int_{B_R} \frac{(\psi^*)^2}{r} \;\leq\; \frac{e^{2R^+\sqrt{d}\,h/s^2}}{(1-\delta)\,\gamma_s}. \tag{32}$$

Outside B_R. By equation 29 and Lemma 3, $r \geq \delta g \geq (\delta/\kappa_G)\psi^*$, so $(\psi^*)^2/r \leq (\kappa_G/\delta)\psi^*$ and

$$\int_{\mathbb{R}^d\setminus B_R} \frac{(\psi^*)^2}{r} \;\leq\; \frac{\kappa_G}{\delta}\,\psi^*(\mathbb{R}^d\setminus B_R). \tag{33}$$

Conclusion. Adding equation 32 and equation 33 and applying Lemma 1 gives

$$\frac{S(r_\theta)}{S(\psi^*)} \;\leq\; \frac{e^{2R^+\sqrt{d}\,h/s^2}}{(1-\delta)\,\gamma_s} \;+\; \frac{\kappa_G}{\delta}\,\psi^*\big(\mathbb{R}^d\setminus B_R\big), \qquad \gamma_s \;\geq\; e^{-L^+ s\sqrt{d}} - ds^2. \tag{34}$$

For the chosen parameters, the second term is at most $(4\kappa_G/\varepsilon)\cdot\varepsilon^2/(16\kappa_G) = \varepsilon/4$. For the first, $\gamma_s^{-1} \leq e^{\varepsilon/8}$ by equation 28 and $e^{2R^+\sqrt{d}\,h/s^2} \leq e^{\varepsilon/8}$ by the choice of h, so it is at most $e^{\varepsilon/4}/(1-\varepsilon/4)$. Using $e^x \leq 1+x+x^2$ on $[0,1]$ with $x = \varepsilon/4 \leq \frac{1}{4}$ gives $e^{\varepsilon/4} \leq 1+\frac{5}{16}\varepsilon$, and $(1+\frac{\varepsilon}{3})(1-\frac{\varepsilon}{4}) = 1+\frac{\varepsilon(1-\varepsilon)}{12} \geq 1$ gives $(1-\varepsilon/4)^{-1} \leq 1+\varepsilon/3$. Hence the first term is at most $(1+\frac{5}{16}\varepsilon)(1+\frac{1}{3}\varepsilon) \leq 1+\frac{3}{4}\varepsilon$ for $\varepsilon \leq 1$. Therefore

$$\frac{S(r_\theta)}{S(\psi^*)} \;\leq\; 1+\tfrac{3}{4}\varepsilon+\tfrac{1}{4}\varepsilon \;=\; 1+\varepsilon,$$

so $\rho(C) \leq 1+\varepsilon$ for $C(\varepsilon) = \lceil 2R^+\sqrt{d}/h\rceil^d + P$. By Lemma 2 the same bound holds for all larger C, and with the lower bound $\rho(C) \geq 1$ of Lemma 1 this proves $\rho \downarrow 1$. □

Proof of Proposition 4. With $q_i = k_i = x_i$ for $i \in [n]$ and $\mathcal{P}_{ij} = 1/n^2$, the joint moment equation 19 factorizes, $Q = B_x^2$ with $B_x(\omega) = \frac{1}{n}\sum_{i=1}^n \phi_\omega^+(x_i)^2$, and completing the square as in Appendix A.1, $p_I(\omega)\phi_\omega^+(x)^2 = e^{\|x\|^2}\mathcal{N}(\omega \mid 2x, I_d)$, so $p_I(\omega)B_x(\omega) \propto \sum_i e^{\|x_i\|^2}\mathcal{N}(\omega \mid 2x_i, I_d)$. Then equation 20 gives $\psi^*(\omega) \propto p_I(\omega)\sqrt{Q(\omega)} = p_I(\omega)B_x(\omega)$, i.e. $\psi^* = \sum_i \pi_i \mathcal{N}(\cdot \mid 2x_i, I_d)$ with $\pi_i \propto e^{\|x_i\|^2}$, a mixture of n Gaussians with covariance I_d. That is a GMProp density equation 6 with $C = n$, means $\mu_i = 2x_i$, unit widths $\sigma_i = 1$ and mixing coefficients $\lambda = \pi$, all strictly positive, so $\psi^* = r_\theta$ for some $\theta \in \Theta(n)$. Hence $S(r_\theta)/S(\psi^*) = 1$ by Lemma 1 and $\rho(n) \leq 1$. With the lower bound $\rho(C) \geq 1$ of Lemma 1, $\rho(n) = 1$. For $C > n$ use Lemma 2. □

Proposition 5 (Covering rate)**.** *Partition $\{x_p\}$ into C groups $\mathcal{C}_1, \ldots, \mathcal{C}_C$ with centres $c_1, \ldots, c_C$, radii $\max_{p\in\mathcal{C}_c}\|x_p - c_c\| \leq \delta$ and weights $W_c = \sum_{p\in\mathcal{C}_c}\bar{\nu}_p$. For every $s > 1$ the mixture*

$$r_C := \sum_{c=1}^{C} \frac{\sqrt{W_c}}{\sum_{c'}\sqrt{W_{c'}}}\,\mathcal{N}(\cdot \mid c_c, s^2 I_d) \in \mathcal{F}(C)$$

satisfies

$$\rho(C) \leq \frac{S(r_C)}{S(\psi^*)} \leq s^d\, e^{\delta^2/(2(s^2-1))}\;\frac{(4\pi)^{d/4}\sum_{c\leq C}\sqrt{W_c}}{Z},$$

and minimizing over s gives $\rho(C) \leq B_C$ of equation 25. The bound is attained when all x_p coincide, with $C = 1$, $\delta = 0$ and $s \downarrow 1$, and the factor $(4\pi)^{d/4}\sum_c\sqrt{W_c}/Z$ tends to 1 when the groups are single points and their separation grows.

Proof. One group. Fix c and $p \in \mathcal{C}_c$, and write $u = \omega - c_c$, $t = \|u\|$. Since $\|\omega - x_p\| \geq (t-\delta)_+$,

$$\frac{\mathcal{N}(\omega \mid x_p, \frac{1}{2}I_d)}{\mathcal{N}(\omega \mid c_c, \frac{s^2}{2}I_d)} = s^d e^{-\|\omega-x_p\|^2+\|u\|^2/s^2} \leq s^d e^{-(t-\delta)_+^2+t^2/s^2}.$$

For $t \geq \delta$ the exponent $-(t-\delta)^2 + t^2/s^2$ is a concave quadratic in t, maximized at $t = \delta s^2/(s^2-1)$ with value $\delta^2/(s^2-1)$; for $t < \delta$ it equals $t^2/s^2 < \delta^2/s^2$, which is smaller. Hence $\mathcal{N}(\cdot \mid x_p, \frac{1}{2}I_d) \leq s^d e^{\delta^2/(s^2-1)} \mathcal{N}(\cdot \mid c_c, \frac{s^2}{2} I_d)$ pointwise, and summing over $p \in \mathcal{C}_c$ with the weights $\bar{\nu}_p$, $h_c := \sum_{p \in \mathcal{C}_c} \bar{\nu}_p \mathcal{N}(\cdot \mid x_p, \frac{1}{2} I_d) \leq W_c \, s^d e^{\delta^2/(s^2-1)} \, \mathcal{N}(\cdot \mid c_c, \frac{s^2}{2} I_d)$.

All groups. $h = \sum_c h_c$, and subadditivity of the root gives

$$\begin{aligned} \sqrt{h} &\leq \sum_c \sqrt{h_c} \leq s^{d/2} e^{\delta^2/(2(s^2-1))} \sum_c \sqrt{W_c} \sqrt{\mathcal{N}(\cdot \mid c_c, \tfrac{s^2}{2} I_d)} \\ &= s^d e^{\delta^2/(2(s^2-1))} \, (4\pi)^{d/4} \sum_c \sqrt{W_c} \mathcal{N}(\cdot \mid c_c, s^2 I_d), \end{aligned}$$

using $\sqrt{\mathcal{N}(\omega \mid c, \frac{s^2}{2} I_d)} = (\pi s^2)^{-d/4} e^{-\|\omega - c\|^2/(2s^2)} = (4\pi)^{d/4} s^{d/2} \mathcal{N}(\omega \mid c, s^2 I_d)$. The right-hand side is $B \, Z \, r_C$ with $B := s^d e^{\delta^2/(2(s^2-1))} (4\pi)^{d/4} \sum_c \sqrt{W_c}/Z$, so $\psi^* = \sqrt{h}/Z \leq B \, r_C$ pointwise.

The ratio. By Lemma 1, $S(r_C)/S(\psi^*) = \int (\psi^*)^2/r_C \leq B \int \psi^* = B$, and $\rho(C) \leq S(r_C)/S(\psi^*)$ since $r_C \in \mathcal{F}(C)$: its weights and width are strictly positive. Minimizing over $s > 1$ gives B_C. If all x_p coincide then $h = W_1 \mathcal{N}(\cdot \mid x_1, \frac{1}{2} I_d)$, $\psi^* = \mathcal{N}(\cdot \mid x_1, I_d)$ and $Z = (4\pi)^{d/4} \sqrt{W_1}$, so with $C = 1$ and $\delta = 0$ the bound tends to $1 = \rho(1)$ as $s \downarrow 1$. If the groups are single points then $h_c = W_c \mathcal{N}(\cdot \mid c_c, \frac{1}{2} I_d)$, and $Z = \int \sqrt{h} \to (4\pi)^{d/4} \sum_c \sqrt{W_c}$ as the centres separate, since the root of a sum of bumps with nearly disjoint supports is nearly the sum of the roots. □

Remark on the implementation. The proofs use only that all mixing coefficients and all widths are strictly positive. The implementation realizes the weights through $\mathrm{softmax}$ and the widths as $e^{\log \sigma}$, both of which land in the open range required above, so no parameter value the optimizer can reach falls outside the family the theorem quantifies over.

B THE DERIVATIVES OF $\log r_\theta$ AT FIXED ω

The first term of equation 17 is $\partial_\theta \hat{\kappa}_\theta(q, k; \omega)|_\omega = -\hat{\kappa}_\theta(q, k; \omega) \, \partial_\theta \log r_\theta$ for every mixture parameter, not only ν. Write $\varphi_c(\omega) = \mathcal{N}(\omega \mid \mu_c, \sigma_c^2 I_d)$ and $\gamma_c(\omega) = \lambda_c \varphi_c(\omega)/r_\theta(\omega)$, the posterior probability of component c. Then $\partial_\theta \log r_\theta = \sum_c \gamma_c \, \partial_\theta \log \varphi_c$ for any parameter of the Gaussians, so

$$\begin{aligned} \frac{\partial \log r_\theta(\omega)}{\partial \mu_c} &= \gamma_c(\omega) \, \sigma_c^{-2} (\omega - \mu_c), \\ \frac{\partial \log r_\theta(\omega)}{\partial \sigma_c} &= \gamma_c(\omega) \, \sigma_c^{-1} \big(\|u_c\|^2 - d \big), \qquad u_c = \sigma_c^{-1}(\omega - \mu_c). \end{aligned} \tag{35}$$

The implemented width parameter is $\log \sigma$, with $\partial/\partial \log \sigma_c = \sigma_c \, \partial/\partial \sigma_c$, hence $\partial \log r_\theta / \partial \log \sigma_c = \gamma_c(\omega) \, (\|u_c\|^2 - d)$. Means and widths of component c affect only φ_c, so both scores carry the single posterior probability γ_c. For the mixing coefficients, $\lambda = \mathrm{softmax}(\nu)$ has $\partial \lambda_{c'}/\partial \nu_c = \lambda_{c'}(\delta_{cc'} - \lambda_c)$, so

$$\frac{\partial \log r_\theta(\omega)}{\partial \nu_c} = \sum_{c'} \frac{\varphi_{c'}(\omega)}{r_\theta(\omega)} \, \lambda_{c'} (\delta_{cc'} - \lambda_c) = \gamma_c(\omega) - \lambda_c \sum_{c'} \gamma_{c'}(\omega) = \gamma_c(\omega) - \lambda_c.$$

Collected, the full gradient of $\log r_\theta$ in θ at fixed ω is

$$\partial_{\nu_c} \log r_\theta = \gamma_c - \lambda_c, \qquad \partial_{\mu_c} \log r_\theta = \gamma_c \, \sigma_c^{-2} (\omega - \mu_c), \qquad \partial_{\log \sigma_c} \log r_\theta = \gamma_c \, (\|u_c\|^2 - d). \tag{36}$$

C THE DEPLOYED FEATURE MAP

The cost tables of Appendix H price one object, which we write down once here. Splitting equation 5 across its two sides in the gauged coordinates of equation 9, deployment with a fixed draw $\omega_1, \ldots, \omega_m \sim r_\theta$ and importance weights $w(\omega_j) = p_I(\omega_j)/r_\theta(\omega_j)$ computes

$$\begin{aligned} \phi_Q(q)_j &= \sqrt{\tfrac{w(\omega_j)}{m}} \, e^{\omega_j^\top A q - \frac{1}{2}\|Aq\|^2}, \\ \phi_K(k)_j &= \sqrt{\tfrac{w(\omega_j)}{m}} \, e^{\omega_j^\top A^{-\top} k - \frac{1}{2}\|A^{-\top} k\|^2}, \end{aligned} \tag{37}$$

so that $\phi_Q(q)^\top \phi_K(k) = \hat{\kappa}_\theta(q,k)$. Read literally, equation 37 makes the gauge look like two extra $d \times d$ maps and two extra quadratic forms per token. Implemented as it should be, one of those four survives, and it is a lower-order term whenever $R \le m$, a few percent in practice (section H.1).

The query self-energy is never evaluated. $-\frac{1}{2}\|Aq\|^2$ is constant in j, so it multiplies the whole vector $\phi_Q(q)$ by one positive scalar. That scalar is a per-*row* factor of $\widehat{K}$, which cancels in the softmax. The deployed query map is therefore

$$\phi_Q(q)_j = \sqrt{\tfrac{w(\omega_j)}{m}}\, e^{\omega_j^\top Aq}, \tag{38}$$

and no quadratic form is ever formed on the query side. The key self-energy is constant in j too, but it is a per-*column* factor and does not cancel: it reweights key j against key j' inside the same row. That single term is the whole residue of the gauge.

ΩA and $\Omega A^{-\top}$ are precomputed. Stack the draw as $\Omega \in \mathbb{R}^{m\times d}$. Neither θ nor A depends on the current tokens (section 3.1), so ΩA and $\Omega A^{-\top}$ are constants of the deployment: they are formed once for each head, feature count m and seed and cached beside the $\frac{1}{2}(\log w(\omega_j) - \log m)$ offsets, and A is never multiplied by again. At run time a query contributes $(\Omega A)q$ and a key $(\Omega A^{-\top})k$, the same single $(L,d)\times(d,m)$ product per side that FAVOR+ pays with Ω alone. This folding is exact for every invertible A, which is why the linear term of equation 37 costs nothing extra rather than merely little.

A rank-R key-side factorization reduces the cost of the surviving term. What is left is $\frac{1}{2}\|A^{-\top}k\|^2 = \frac{1}{2}k^\top(A^{-1}A^{-\top})k$, a dense $d \times d$ form per key that no precomputation removes: the matrix is constant but the vector it acts on is the token. We therefore parameterize the key-side map itself in factored form,

$$A^{-\top} = O\big(\Delta + UV^\top\big), \qquad \Delta \text{ a positive diagonal}, \quad U,V \in \mathbb{R}^{d\times R}, \quad O \in \mathrm{SO}(d), \tag{39}$$

which recovers a dense A at $R = d$. Writing $B = \Delta + UV^\top$, the matrix determinant lemma gives $\det B = \det\Delta\, \det(I_R + V^\top\Delta^{-1}U)$, so with Δ positive the map is invertible exactly when the $R \times R$ core $I_R + V^\top\Delta^{-1}U$ is. That core is the same matrix the Woodbury solve below forms, and a step approaching singularity registers in the gauge's condition number, which the run tracks. Because O is orthogonal it leaves the norm alone,

$$\big\|A^{-\top}k\big\|^2 = \big\|\Delta k + U(V^\top k)\big\|^2, \tag{40}$$

so the self-energy costs $O(dR)$ per token rather than $O(d^2)$, and O appears only in the linear term, where it has already been folded into $\Omega A^{-\top}$: an unconstrained rotation composed with a low-cost stretch. Both directions stay closed form: $A^{-\top} = OB$ and $A = B^{-\top}O^\top$ costs one $R\times R$ Woodbury solve, so nothing on the export path takes a $d \times d$ inverse, square root or eigendecomposition. At deployment the gauge reduces to the $d(2R{+}1)$ floats of (Δ, U, V), which could be collapsed to the d^2 of $\Delta + UV^\top$ at $R = d$, though the reported runs keep the factored form (Appendix H). O and A itself are absorbed into the two cached projections.

D Direct variance minimization: a negative result

GMProp is placed, in section 5, inside adaptive importance sampling, section 3.4 defines the object variance-minimizing adaptive importance sampling targets and Appendix A derives its integral form: the second moment $S(\psi) = \frac{1}{m}\int f^2/\psi$ of equation 19, with $f(\omega) = p_I(\omega)\sqrt{Q(\omega)}$, minimized by $\psi^* \propto f$. Below we write Q in the form it takes under uniform weights over all pairs of $\mathcal{D}$, $Q = B_qB_k$ with $B_q(\omega) = \frac{1}{N}\sum_{i\in[N]}\phi^+_\omega(q_i)^2$ and B_k likewise, for readability. Nothing below uses the factorization, and with the pair weights of Theorem 1 one reads Q for B_qB_k throughout. A family that provably approaches ψ^* (Theorem 2) suggests training it on S directly. This training fails, and this appendix records how, because the failure is specific enough to be worth reporting: the objective is not merely hard to optimize, it is optimizable in a direction that makes the estimator worse.

Four things go wrong and we discuss about these in the following . The pathwise gradient of the self-sampled loss has a minimum that is easy to reach and is not a better proposal. The score-function

and off-policy versions of the objective change the proposal only slightly and never close the gap to p_I. S is an unnormalized second moment, so it does not even rank heads the way accuracy does. And S is a functional of the marginal alone, so it cannot see the coupling of the m draws. Only the first is a bug in how we trained.

The objective and its two estimators. Since $f(\omega) = p_I(\omega)\sqrt{B_q(\omega)B_k(\omega)}$, the same integral can be written as an expectation under the proposal or under the isotropic law,

$$m\,S(r_\theta) = \int \frac{p_I^2 B_q B_k}{r_\theta} = \mathbb{E}_{\omega\sim r_\theta}\Big[\Big(\tfrac{p_I}{r_\theta}\Big)^2 B_q B_k\Big] = \mathbb{E}_{\omega\sim\mathcal{N}(0,I_d)}\Big[\tfrac{p_I}{r_\theta} B_q B_k\Big], \tag{41}$$

and more generally under any μ with the weight $p_I^2/(r_\theta\mu)$. The first form is the natural one to train: ω comes from equation 14, so the gradient is the pathwise one of equation 17 and nothing extra is needed. It is also the one that fails. The two forms are estimators of a single number, so we report both, and their disagreement is the diagnostic that the rest of this section turns on.

Three ways to differentiate it. Write $M_2 = m\,S(r_\theta) = \int f^2/r_\theta$ for the integral itself and let μ be the law the m training draws come from, so that $\mu = r_\theta$ on-policy. Draw s contributes $w_s^2\,B_q(\omega_s)B_k(\omega_s)$ to the estimate, with the weight $w_s = p_I(\omega_s)/\sqrt{r_\theta(\omega_s)\,\mu(\omega_s)}$ that makes the average converge to M_2 from μ, and it equals p_I/r_θ only when $\mu = r_\theta$. Write ℓ_s for the log of that contribution, so $\log\widehat{M_2} = \log\sum_s e^{\ell_s} - \log m$ and $\mathrm{softmax}(\ell)_s$ is draw s's share of the estimate. Training descends $\log\widehat{M_2}$. Differentiating the integral needs no Monte Carlo at all:

$$\begin{aligned}\nabla_\theta M_2 &= \nabla_\theta \int f^2\, r_\theta^{-1}\,\mathrm{d}\omega = \int f^2\,\nabla_\theta\big(r_\theta^{-1}\big)\,\mathrm{d}\omega = -\int f^2\, r_\theta^{-2}\,\nabla_\theta r_\theta\,\mathrm{d}\omega \\ &= -\int r_\theta\Big(\frac{f}{r_\theta}\Big)^2\nabla_\theta\log r_\theta\,\mathrm{d}\omega = -\,\mathbb{E}_{\omega\sim r_\theta}\Big[\underbrace{\Big(\tfrac{p_I}{r_\theta}\Big)^2 B_q B_k}_{\text{weight}}\ \underbrace{\nabla_\theta\log r_\theta}_{\text{score}}\Big],\end{aligned} \tag{42}$$

so $\nabla_\theta\log M_2 = -\mathbb{E}_{\omega\sim\pi_\theta}[\nabla_\theta\log r_\theta]$ with $\pi_\theta \propto f^2/r_\theta$, the law of where the integral's mass lies. The three configurations differ only in which finite-sample version of this they hand the optimizer.

On-policy, pathwise. The sampler is r_θ itself, and equation 14 pushes its θ-dependence into the sample, $\omega = \omega_\theta(z;c)$ with $c\sim\mathrm{Cat}(\lambda)$ detached and $z\sim\mathcal{N}(0,I_d)$. The law of z does not depend on θ and that of c is detached, so for μ,σ,A the integrand carries all of the θ-dependence, and with $h(\theta,\omega) = \big(p_I(\omega)/r_\theta(\omega)\big)^2 B_q(\omega)B_k(\omega)$,

$$M_2 = \mathbb{E}_{c,z}\big[h\big(\theta,\omega_\theta(z;c)\big)\big], \qquad \nabla_\theta M_2 = \mathbb{E}_{c,z}\Big[\underbrace{-2h\,\nabla_\theta\log r_\theta}_{\text{through the weight}} + \underbrace{(\nabla_\omega h)^\top\frac{\mathrm{d}\omega}{\mathrm{d}\theta}}_{\text{through the samples}}\Big], \tag{43}$$

the two routes of equation 17. It is exact in μ,σ,A, while in ν it omits the term through the index's law, $\mathbb{E}_{c,z}[h\,\nabla_\nu\log\lambda_c]$, which section 3.5 drops by detaching c. Only this configuration has the second term. It is a derivative with respect to *where the draws are*, and the finite sum it acts on cannot report the mass those draws are shifted away from.

On-policy, score-function. The draws are the same and are detached. What the optimizer gets is the self-normalized estimator of $\nabla_\theta\log M_2$ that equation 42 licenses, with $\mathrm{softmax}(\ell)$ standing in for π_θ:

$$\widehat{\nabla_\theta\log M_2} = -\sum_{s=1}^{m}\mathrm{softmax}(\ell)_s\;\nabla_\theta\log r_\theta(\omega_s). \tag{44}$$

Nothing is differentiated but $\log r_\theta$ at fixed points, so there is no route through $\mathrm{d}\omega/\mathrm{d}\theta$ and no analogue of the second term of equation 43. Carrying the categorical term explicitly also restores the score in ν that the detached index of equation 14 drops, so nothing is omitted here.

Off-policy. The draws come from a μ that does not depend on θ, so $\mathrm{d}\omega/\mathrm{d}\theta = 0$ and the second term of equation 43 vanishes identically. What is left is not an approximation of equation 44. It is the

same expression, reached by differentiating $\log \widehat{M}_2$ directly:

$$\begin{aligned}\nabla_\theta \log \widehat{M}_2 &= \frac{1}{\widehat{M}_2}\, \nabla_\theta \frac{1}{m}\sum_{s=1}^{m} \frac{f^2(\omega_s)}{\mu(\omega_s)}\, r_\theta(\omega_s)^{-1} \\ &= -\frac{\sum_s \dfrac{f^2(\omega_s)}{r_\theta(\omega_s)\mu(\omega_s)}\, \nabla_\theta \log r_\theta(\omega_s)}{\sum_t \dfrac{f^2(\omega_t)}{r_\theta(\omega_t)\mu(\omega_t)}} = -\sum_{s=1}^{m} \mathrm{softmax}(\ell)_s\, \nabla_\theta \log r_\theta(\omega_s).\end{aligned} \tag{45}$$

Off-policy, then, the pathwise and score-function settings are one map, which is why that configuration is one row of Table 6 and not two.

Why the self-sampled estimate can fall without a better proposal, and how the table detects it. The on-policy pathwise configuration closes a loop the other two leave open. Here r_θ supplies the draws, the pathwise gradient shifts those same draws, and r_θ also appears in the denominator of the integrand. The integral $M_2 = \int f^2/r_\theta$ is governed by the regions where r_θ is small, which are exactly the regions that draws $\omega \sim r_\theta$ rarely land in, so a finite self-sampled estimate $\widehat{M}_2$ can miss the regions that contribute most to M_2. The second term of equation 43 is a derivative with respect to *where the draws are*, and it lowers $\widehat{M}_2$ by shifting them away from a high-contribution region: the realized sample then no longer represents that region, while the estimand M_2 still carries its contribution in full. That is a spurious finite-sample descent direction. Covering ψ^* better also lowers $\widehat{M}_2$, but withdrawing from the regions that matter is easier to reach, and along it $\widehat{M}_2$ falls while the true M_2 rises by orders of magnitude, as the table shows.

The independent estimator of equation 41 is the detector. Its draws come from $\mathcal{N}(0, I_d)$, whose law does not depend on θ, so no region can be vacated unobserved, and the two readings estimate one number. Both are logs of Monte Carlo averages and so fall below the truth by Jensen's inequality, the more so the heavier the sampler's weight tail. The *larger* reading is therefore the reliable one, and a self-sampled reading far below its independent partner is not a better proposal but a sampler that has left the mass behind. This is the concrete reason variance-minimizing adaptive importance sampling scores proposals on held-out draws rather than on the sampler's own.

Protocol. Pythia-70M, layer 0, all 8 heads fit independently on the pile mix, gauge off, C=16 means compressed from 262144 pair-sum candidates by 25 iterations of k-means. Each of 2000 training steps draws m=128 features and resamples 128 of 2000 training contexts. The loss is the mean over contexts of $\log \widehat{M}_2$. Selection is on validation $\log S$ at m=64 over 340 contexts. The reported figure is $\log(mS) = \log M_2$, which by Proposition 3 does not depend on m, on 512 held-out test contexts at $m \in \{16, 256, 4096\}$ over seeds $\{0, 1, 2, 3, 4\}$, mean $\pm$ std over the 8 heads, together with the log-kernel MSE on the same test block as mean $\pm$ std over heads $\times$ seeds. Arithmetic is float64. Each on-policy configuration is trained twice, with equation 43 and with equation 44. Off-policy the two coincide, so that configuration is one row.

- *On-policy, initialization.* $\omega \sim r_\theta$, with r_θ initialized at the r_0 of Algorithm 2.
- *On-policy, FAVOR+ initialization.* The same loss from the neutral initialization of Proposition 2, where $r_\theta = p_I$ exactly. This is the unambiguous test: the run is initialized *at* the baseline it is compared against, so any change is attributable to training.
- *Off-policy.* ω is drawn from a fixed snapshot of r_0 with no gradient through the draw, so the sampler cannot follow θ and only $\log r_\theta$ is differentiated. A 5% share of an equal-weight copy of the initial components is mixed defensively into r_θ, and selection is made on draws from $\mathcal{N}(0, I_d)$.

The reference row is FAVOR+, the untrained $r = p_I$.

The pathwise gradient of the self-sampled loss has a minimum that does not improve the proposal. In the row of Table 6 for the pathwise run initialized at FAVOR+, its own training signal reports $\log(mS) = -2.84\times10^7$ at m=16 against FAVOR+'s -59.7: an improvement of 2.8×10^7 nats, by a run initialized exactly at FAVOR+. The independent estimator of the same integral reports

Table 6: Layer-0 $\log(mS)$ ($\downarrow$, nats) of proposals trained directly on the second moment, Pythia-70M on the pile mix, 512 held-out contexts, mean $\pm$ std over the 8 layer-0 heads. The two estimator rows are equation 41 on the same θ: one formula, $\log\left(\frac{1}{m}\sum_{s\le m} w_s^2\, B_q(\omega_s)B_k(\omega_s)\right)$, each under its own (ω, w), either $\omega \sim \mathcal{N}(0, I_d)$ with $w = \sqrt{p_I/r_\theta}$ or $\omega \sim r_\theta$ with $w = p_I/r_\theta$. Both estimate $\int f^2/r_\theta$. A gap between them is not sampling noise: the larger reading is the reliable one, and a self-sampled reading far below its independent partner is a sampler that has withdrawn from the mass. The last three columns are the log-kernel MSE ($\downarrow$, nats^2) of Appendix E on the same contexts over seeds $\{0, \ldots, 4\}$, mean $\pm$ std over heads $\times$ seeds, which the objective never sees. FAVOR+ is untrained, so its two estimators coincide. The off-policy row is a single run, because its two gradients coincide by equation 45. The last row is not trained on S: it is the proposal of Table 1 trained on $\mathcal{L}_{\log}$, scored under the identical protocol, so its log-kernel columns reproduce the means of that table's Pythia-70M with-gauge cells. †That proposal carries a learned gauge and S is not gauge-invariant: A leaves K_{ij} unchanged but changes B_q and B_k (Appendix A.3), so its $\log(mS)$ is a different integral from the ungauged rows and should not be read straight down the column.

		$\log(mS)$ at $m =$			log-kernel MSE at $m =$		
Run	Estimator	16	256	4096	16	256	4096
FAVOR+ ($r = p_I$, untrained)	either	-59.7 ± 76.6	-40.9 ± 64.7	-27.4 ± 60.7	$2.91\times10^{3} \pm 3.88\times10^{3}$	$1.76\times10^{3} \pm 2.47\times10^{3}$	$1.26\times10^{3} \pm 1.96\times10^{3}$
On-policy, initialization r_0 (pathwise)	independent ($\omega \sim \mathcal{N}(0, I_d)$, $w = \sqrt{p_I/r_\theta}$)	$2.27\times10^{16} \pm 2.44\times10^{16}$	$2.29\times10^{16} \pm 2.46\times10^{16}$	$2.30\times10^{16} \pm 2.47\times10^{16}$	$2.77\times10^{9} \pm 1.13\times10^{9}$	$2.76\times10^{9} \pm 1.13\times10^{9}$	$2.76\times10^{9} \pm 1.13\times10^{9}$
	self-sampled ($\omega \sim r_\theta$, $w = p_I/r_\theta$)	$-1.01\times10^{5} \pm 2.44\times10^{4}$	$-1.01\times10^{5} \pm 2.44\times10^{4}$	$-1.01\times10^{5} \pm 2.44\times10^{4}$			
On-policy, initialization r_0 (score-fn)	independent ($\omega \sim \mathcal{N}(0, I_d)$, $w = \sqrt{p_I/r_\theta}$)	2.38 ± 38.8	11.1 ± 35.8	17.8 ± 34.4	805 ± 1136	391 ± 624	236 ± 405
	self-sampled ($\omega \sim r_\theta$, $w = p_I/r_\theta$)	31.2 ± 15.9	39.3 ± 15.8	44.3 ± 16.6			
On-policy, FAVOR+ initialization (pathwise)	independent ($\omega \sim \mathcal{N}(0, I_d)$, $w = \sqrt{p_I/r_\theta}$)	$2.63\times10^{8} \pm 1.62\times10^{8}$	$2.63\times10^{8} \pm 1.62\times10^{8}$	$2.64\times10^{8} \pm 1.62\times10^{8}$	$2.11\times10^{14} \pm 8.88\times10^{13}$	$2.11\times10^{14} \pm 8.88\times10^{13}$	$2.09\times10^{14} \pm 8.75\times10^{13}$
	self-sampled ($\omega \sim r_\theta$, $w = p_I/r_\theta$)	$-2.84\times10^{7} \pm 6.63\times10^{6}$	$-2.84\times10^{7} \pm 6.63\times10^{6}$	$-2.83\times10^{7} \pm 6.52\times10^{6}$			
On-policy, FAVOR+ initialization (score-fn)	independent ($\omega \sim \mathcal{N}(0, I_d)$, $w = \sqrt{p_I/r_\theta}$)	-59.7 ± 76.6	-40.9 ± 64.7	-27.4 ± 60.7	$4.46\times10^{3} \pm 6.92\times10^{3}$	$3.11\times10^{3} \pm 4.94\times10^{3}$	$2.41\times10^{3} \pm 4.05\times10^{3}$
	self-sampled ($\omega \sim r_\theta$, $w = p_I/r_\theta$)	-59.7 ± 76.6	-40.9 ± 64.7	-27.4 ± 60.7			
Off-policy, fixed r_0, ε=0.0500 (score-fn ≡ pathwise)	independent ($\omega \sim \mathcal{N}(0, I_d)$, $w = \sqrt{p_I/r_\theta}$)	-10.3 ± 50.0	-1.16 ± 42.2	6.44 ± 39.7	461 ± 608	230 ± 340	130 ± 208
	self-sampled ($\omega \sim r_\theta$, $w = p_I/r_\theta$)	36.5 ± 14.9	42.1 ± 15.8	46.4 ± 17.1			
Trained on $\mathcal{L}_{\log}$ (Table 1, gauge)†	independent ($\omega \sim \mathcal{N}(0, I_d)$, $w = \sqrt{p_I/r_\theta}$)	-22.8 ± 48.3	-13.0 ± 40.5	-6.68 ± 38.6	69.4 ± 53.3	23.0 ± 20.4	9.14 ± 8.97
	self-sampled ($\omega \sim r_\theta$, $w = p_I/r_\theta$)	34.0 ± 11.4	40.3 ± 13.2	45.1 ± 15.2			

$+2.63\times10^{8}$. The two disagree by 2.92×10^{8} nats on average and by 4.93×10^{8} on the worst head. Validation $\log S$ fell by between 2.0×10^{7} and 3.6×10^{7} nats per head while log-kernel MSE rose on every one of the eight, $2.91\times10^{3} \to 2.11\times10^{14}$ in the mean at m=16. The pathwise initialization row does the same thing from r_0: self-sampled -1.01×10^{5} against an independent reading of 2.27×10^{16}. That is the withdrawal mechanism above, read off the table.

Which of the two routes to θ finds it is the part the paired rows pin down, and the answer is that self-sampling alone is not the fault. The score-function gradient descends the identical loss on identical draws and does not collapse: from r_0 it changes validation $\log S$ by at most 1.2 nats and five of the eight heads select step 0. Initialized at FAVOR+, all eight do, and the published $\log(mS)$ is bit-identical to the untrained row. Only the pathwise route reaches the withdrawal direction, because withdrawal acts on *where the samples are*: it is the second term of equation 43, and equation 44 has no term of that kind. The gradient route, not the estimator, is what has to be fixed.

Removing the circularity removes the collapse and changes nothing else. Drawing the training points from a fixed law does what it was meant to do. The estimator gap at m=16 falls from $+2.92\times10^{8}$ nats on the pathwise run initialized at FAVOR+ to -47, and its sign flips: the self-sampled number is now the conservative one, which is what should happen once the sampler no longer follows θ. Validation $\log S$ changes by 1.2 to 52 nats per head instead of tens of millions, and selection lands mid-run rather than at the last step. The off-policy and score-function routes are mechanically different, since one fixes the sampler and the other keeps it and changes the gradient, and they agree, which is what makes the rest of this section a statement about the objective and the family rather than about the optimizer.

It also brings no gain on S. The off-policy run, a single row because its two gradients coincide, reaches $36.5/42.1/46.4$ against the on-policy score-function initialization run's $31.2/39.3/44.3$, both self-sampled as printed in Table 6. Taking the pessimistic estimator per head before averaging over heads puts them at $36.5/42.1/46.4$ and $35.8/42.9/47.1$, so the two runs end in the same place either way. Both lose to FAVOR+ on all eight heads at every feature count, by 96, 83 and 74 nats in the mean at m=16, 256 and 4096. Per head that gap runs from 10 to 271 nats, median near 60, and two heads carry the mean, as shown below. The circularity was a real bug in the pathwise training signal and not the reason the objective does not deliver.

Why the mixture loses on S even in the runs without withdrawal. S is a χ^2 (Lemma 1), so it is governed by the worst-covered region rather than by the average fit, and it punishes a proposal for being small somewhere far more than it rewards it for being well-placed everywhere else. FAVOR+

is the one candidate here that by construction cannot be small anywhere: $r = p_I$ has no holes. A 16-component mixture on compressed pair sums has many, and the compression makes them worse than the theory suggests. The G-weighted effective sample size of the 262144-pair candidate block was between 3.1 and 8879 across the eight heads, below 60 on four of them, so most of the 16 components are initialized on essentially no mass. Neither remedy we used reaches those gaps. The fixed sampler draws from r_0, whose support is the components themselves, so the space between them never enters the gradient. And ε-mixing r_0 floors r_θ against r_0, which has the same holes, rather than against p_I, which does not.

And S is not the quantity the paper is about. $S = \frac{1}{m}\int f^2/r$ is an *unnormalized* second moment: $f \mapsto cf$ sends $S \mapsto c^2 S$. Within one head that is harmless, because $\int f$ does not depend on θ and minimizing S is exactly minimizing the estimator's relative variance, but the scale of f varies by hundreds of nats across the heads of one layer, so a tiny S can mean features that are silent rather than accurate. On head 3 FAVOR+ reaches $\log(mS) = -232$ at m=16 because $\mathbb{E}_{\omega\sim\mathcal{N}(0,I_d)}[B_q B_k]$ is nearly zero there, and the same isotropic law loses to every GMProp row and to FAVOR# in every column of Table 1. Across the eight heads FAVOR+'s $\log(mS)$ and its log-kernel MSE are rank-reversed, with Spearman $\rho = -1.00$, -0.95 and -0.98 at m=16, 256 and 4096 and Pearson -1.00 to two places at all three. Its best head on the objective, head 3, is its worst on the log-kernel MSE, at 1.21×10^4 for m=16, and its worst, head 0 at $+1.1$, is its best, at 1.28×10^2. A mean of $\log(mS)$ over heads therefore reports feature scale, not coverage of ψ^*. It is why the mean gap to FAVOR+ above is $1.6\times$ its median, carried by heads 3 and 1, and why only the per-head statement, for every head and every m, survives the normalization, the averaged nat counts being descriptive of this layer rather than an effect size. Training against the exact kernel has no such degeneracy, which is why the S runs without withdrawal improve log-kernel MSE at m=16 from 2.91×10^3 to 461 off-policy and to 805 on-policy with the score-function gradient, while still losing on S, why section 3.4 trains on the kernel, and why the χ^2 geometry enters this paper only through Theorem 2, as a statement about what the family can represent rather than about what an optimizer will find.

And S cannot see the coupling, which accounts for FAVOR+'s remaining margin. S is a functional of the marginal r alone. It says nothing about the joint law of the m draws, which is the object the coupling literature of section 5 works on, so two estimators that draw the same marginal differently are the same point of this objective. The score-function row initialized at FAVOR+ measures what that costs. Every head selects step 0, so $r_\theta = p_I$ identically and its $\log(mS)$ is bit-identical to the untrained row at all three feature counts. Its log-kernel MSE is nonetheless $4.46\times10^3/3.11\times10^3/2.41\times10^3$ against FAVOR+'s $2.91\times10^3/1.76\times10^3/1.26\times10^3$. The same density, 1.5 to $1.9\times$ the error, because FAVOR+ draws an orthogonal block while the proposal path draws its m points independently.

Two things follow. A proposal trained on S could match p_I exactly and still lose at finite m, so S is not a sufficient target even where its estimate is reliable. And the log-kernel columns of Table 6 give FAVOR+ an orthogonality the rows trained on S do not have: the runs without withdrawal surpass it there by 3.6 to $9.7\times$ anyway, carrying the handicap that the score-function row initialized at FAVOR+ measures.

What replaces S is also a variance. None of the above abandons variance minimization. By Proposition 1, $\mathbb{E}_{\omega_{1:m}\sim r_\theta}[\widehat{K}_{ij}] = K_{ij}$ at every θ, so squared error against the exact kernel is the estimator's own variance with nothing assumed about the fit (equation 11), and averaged over pairs it is S again: Appendix A records the expected Monte Carlo variance of one term as $m\,S(\psi)$ minus the constant $\mathbb{E}_{q,k}e^{2q^\top k}$, which does not depend on the proposal. The objective of Table 1 and the objective of this appendix are the same functional of the same estimator. What differs is the weighting and the reference, and every failure above comes from these two. The squared error is taken per pair and, in the log coordinate of Appendix E, divided by that pair's own K_{ij}^2, so the average is a *relative* variance. That is the normalization S lacks, and it removes the head-scale artifact. And K_{ij} is computed exactly from the recorded q_i, k_j rather than estimated from the proposal's own draws, so there is no sampling design for θ to exploit, no region that can be vacated unobserved, and an estimator that goes silent is maximally wrong instead of maximally rewarded. Neither fix touches the family: r_θ is the same object either way, and only what is asked of it changes.

Same minimizer, different noise. Variance-minimizing adaptive importance sampling trains on the second moment $\mathbb{E}_{\omega\sim r_\theta}[\hat\kappa^2]$ rather than on $\mathbb{E}_{\omega\sim r_\theta}[(\hat\kappa-\kappa)^2]$, with $\hat\kappa=\hat\kappa_\theta(q,k;\omega)$ the estimator equation 10 at a single draw and $\kappa=\kappa(q,k)$ the exact kernel of the pair, and at the population level nothing separates them: $\mathbb{E}_{\omega\sim r_\theta}[(\hat\kappa-\kappa)^2]=\mathbb{E}_{\omega\sim r_\theta}[\hat\kappa^2]-\kappa^2$ with κ independent of θ, so the minimizers coincide. Their Monte Carlo estimates are not interchangeable. For that pair, let $\hat\kappa_r=\hat\kappa_\theta(q,k;\omega_r)$ with $\omega_r\sim r_\theta$ i.i.d., $r=1,\dots,R$, so $\hat\kappa_r\ge 0$ and $\mathbb{E}_{\omega_r\sim r_\theta}[\hat\kappa_r]=\kappa$ (Proposition 1), and compare the two unbiased estimates $\frac{1}{R}\sum_r\hat\kappa_r^2$ and $\frac{1}{R}\sum_r(\hat\kappa_r-\kappa)^2=\frac{1}{R}\sum_r\hat\kappa_r^2-2\kappa(\bar\kappa_R-\kappa)-\kappa^2$, with $\bar\kappa_R=\frac{1}{R}\sum_r\hat\kappa_r$. The second is the first with a control variate, $2\kappa(\bar\kappa_R-\kappa)$, of mean zero because κ is known exactly, and it is always a reduction: with $\xi=\hat\kappa-\kappa$, $\hat\kappa^2=\xi^2+2\kappa\xi+\kappa^2$ gives, with every variance, covariance and expectation over $\omega\sim r_\theta$, $\mathrm{Var}_\omega(\hat\kappa^2)-\mathrm{Var}_\omega(\xi^2)=4\kappa^2\mathrm{Var}_\omega(\xi)+4\kappa\,\mathrm{Cov}_\omega(\xi^2,\xi)=4\kappa\,\mathbb{E}_\omega[\xi^2(\kappa+\xi)]$, so

$$\mathrm{Var}_{\omega_{1:R}}\Big(\frac{1}{R}\sum_r\hat\kappa_r^2\Big)-\mathrm{Var}_{\omega_{1:R}}\Big(\frac{1}{R}\sum_r(\hat\kappa_r-\kappa)^2\Big)=\frac{4\kappa}{R}\,\mathbb{E}_\omega\big[\hat\kappa\,(\hat\kappa-\kappa)^2\big]\;\ge\;0, \tag{46}$$

with equality only when $\hat\kappa=\kappa$ almost surely, i.e. when the proposal is already exact and both variances vanish. Nothing but $\hat\kappa\ge 0$ and $\mathbb{E}_\omega[\hat\kappa]=\kappa$ was used, so the statement holds verbatim with the m-draw estimator $\hat\kappa_\theta(q,k)$ in place of $\hat\kappa$. It is a statement about the loss estimate: differentiated, the same term enters the pathwise gradient of equation 17 as the zero-mean $2\kappa\,\nabla_\theta\bar\kappa_R$, and we claim no inequality for the gradient. The reference the optimizer cannot change is thus also a control variate available at no cost, which the unnormalized S of this appendix never had.

What we did not try. Four further changes are not covered by the runs above: sampling the training loss from $\mathcal{N}(0,I_d)$ itself, or from a mixture floored by p_I, so that the gaps do enter the gradient; a defensive component anchored on p_I rather than on r_0; wider or fewer components, since the effective sample sizes above say the 16-component compression is being asked to place more mass than the block contains; and coupling the m draws, which the implementation already supports as orthogonal or simplex blocks. We report the negative result rather than a tuned version of it because the conclusion we draw, to train on a target the optimizer cannot change, does not depend on which of those would have closed the gap.

Reproducing it. Every run is the procedure of the $\mathcal{L}_{\log}$ rows in Appendix G with $\log\widehat{M_2}$ in place of the log-kernel MSE. The three sampler configurations are the self-sampled run from r_0, the self-sampled run from the neutral FAVOR+ initialization of Proposition 2, and the off-policy run, which draws from a fixed copy of r_0, mixes 5% of an equal-weight copy of the initial components into r_θ and selects on draws from $\mathcal{N}(0,I_d)$. Each on-policy configuration is run once with the pathwise gradient equation 43 and once with the score-function gradient equation 44. Off-policy the two coincide equation 45, so that configuration is one run. All five trained runs draw uncoupled features, against the orthogonal block of the FAVOR+ reference in the log-kernel columns, while its $\log(mS)$ row draws i.i.d. Gaussians. The last row draws as Table 1 does, with stratified components and orthogonal blocks. Both estimators of equation 41 are reported at every test m, together with their gap.

E THE LOG-KERNEL MSE

Notation. The tables report $\mathcal{L}_{\log}$ of equation 12 with the pair average taken over the causal pairs $j\le i$ of the scored contexts, the pair law $\mathcal{P}$ of section 2, and over the feature-draw seeds. Entrywise,

$$\mathbb{E}_{(i,j)\sim\mathcal{P},\,\omega_{1:m}}\big[(\log\widehat{K}_{ij}-\log K_{ij})^2\big]\;:=\;\mathrm{mean}_{ij}\,\big(\log\widehat{K}_{ij}-\log K_{ij}\big)^2, \tag{47}$$

the mean of the squared log-error over those pairs, contexts and seeds. The expectation is over the pair and the draw and is estimated by exactly that average.

What it stands in for. Write $\delta_{ij}=(\widehat{K}_{ij}-K_{ij})/K_{ij}$ for the estimator's relative error at one pair. Then

$$\log\widehat{K}_{ij}-\log K_{ij}=\log\big(1+\delta_{ij}\big)\;\approx\;\delta_{ij}=\frac{\widehat{K}_{ij}-K_{ij}}{K_{ij}}, \tag{48}$$

and squaring gives equation 49 below. By Proposition 1, $\mathbb{E}_{\omega_{1:m}\sim r_\theta}[\widehat{K}_{ij}] = K_{ij}$, which makes the numerator a variance rather than a mean squared error around another value:

$$\mathbb{E}_{\omega_{1:m}}\big[\big(\log \widehat{K}_{ij} - \log K_{ij}\big)^2\big] \;\approx\; \frac{\mathrm{Var}_{\omega_{1:m}}\big(\widehat{K}_{ij}\big)}{K_{ij}^2}. \tag{49}$$

The reported number is therefore a stand-in for the *relative* variance of the kernel estimator, which is the scale-invariant quantity such a replacement should be judged on.

Why the surrogate and not the quantity itself. equation 49 is used left to right, not right to left: we report the log error because the relative variance on its right is not measurable on these models. The kernel entries $K_{ij} = e^{q_i^\top k_j}$ span a large part of the float range within a single head, with a few pairs near the top of it and most of the mask effectively zero, and any average of $(\widehat{K}_{ij} - K_{ij})^2$, normalized or not, inherits that spread squared.

Two failures follow. Where K_{ij} is small the estimate underflows to zero, the pair contributes exactly K_{ij}^2 however good the proposal is, and its summand stops depending on m altogether. As a training signal that region is flat, and a run descending it stalls. Where K_{ij} is large a handful of pairs dominate the average, so the statistic reports the corpus's dynamic range rather than the estimator's accuracy.

Taking the log first removes both failures at once, because $\log K_{ij} = q_i^\top k_j$ is a quantity of ordinary size. Every admitted pair then carries comparable weight, a pair the estimator misses becomes a large but finite penalty instead of a zero or a `NaN`, and the score changes with the feature count: the FAVOR+ row of Table 7 on Pythia-70M reads $2.91{\times}10^3$, $1.76{\times}10^3$ and $1.26{\times}10^3$ at m=16, 256 and 4096.

The cost of that choice is that equation 48 is a first-order expansion, exact only as $\delta_{ij} \to 0$, so equation 49 holds where $\widehat{K}_{ij}$ is already close to K_{ij} and degrades where it is not. In several cells of Table 1, the untrained baselines at m=16 especially, it is not. Every table in this paper should therefore be read as what it literally reports, squared error on the logs, which is the quantity we can compute. $\mathrm{Var}_{\omega_{1:m}}(\widehat{K}_{ij})/K_{ij}^2$ is what it stands in for, not a number the tables contain.

F Corpora

The two pretraining-mixture tables of the main text (Table 1, Table 2) fit and score on a sample of the model's own pretraining corpus, not on a corpus of a single genre: Dolma v1.5 at its published token shares for OLMo-1B, and for Pythia a genre-balanced sample of the copyright-stripped Pile release. Each mixture is divided by document into training, validation and test sets of 90, 5 and 5 percent. A context is the first 512 tokens of an excerpt, or the whole excerpt if it is shorter, and section 4 gives the number of tokens in each set. The cross-corpus test of section 4.3 scores the same proposals on a third corpus, described last.

Pile mix for Pythia. Seventeen components drawn from `monology/pile-uncopyrighted`, the copyright-stripped Pile, which removes Books3, BookCorpus2, OpenSubtitles, YTSubtitles and OWT2. Shares are rebalanced relative to the official mix, with six small domains floored at 1%: Pile-CC 22%, Gutenberg (PG-19) 9%, ArXiv 9%, Github 9%, Wikipedia (en) 7%, StackExchange 7%, PubMed Central 6%, Ubuntu IRC 6%, FreeLaw 5%, HackerNews 4%, Enron Emails 4%, USPTO Backgrounds 4%, PubMed Abstracts 3%, DM Mathematics 2%, EuroParl 1%, PhilPapers 1%, NIH ExPorter 1%. The scored set is 82,751 training and 4,628 test excerpts, with a fixed 340-excerpt subsample of the 5% validation set used for proposal selection. This is a genre-balanced sample of the corpus Pythia was trained on (Gao et al., 2020; Biderman et al., 2023), not the official pretraining marginal.

Dolma mix for OLMo-1B. Dolma v1.5 at its published token shares (Soldaini et al., 2024; Groeneveld et al., 2024): Common Crawl ≈ 75%, split over the tail, middle and head quality tiers as 38.7%, 20.0% and 15.8%, The Stack 13.4%, C4 6.5%, Reddit 2.9%, peS2o 2.3%, books 0.20%, Wikipedia 0.14%. The scored set is 105,860 training and 5,895 test excerpts, with a fixed 155-

excerpt subsample of the 5% validation set used for selection. Unlike the pile mix, the shares are corpus-native: this is the distribution `allenai/OLMo-1B-hf` was pretrained on.

UN Parallel Corpus. The United Nations Parallel Corpus v1.0 (Ziemski et al., 2016), which the cross-corpus test of section 4.3 uses, is a corpus of official UN documents of 1990–2014 in the public domain, such as resolutions, reports and committee records, manually translated into the six official languages and sentence-aligned for machine translation. We take the English side of the OPUS raw release (Tiedemann, 2012), one TEI XML file per document, $159{,}336$ documents in all, and rebuild each document's text from its paragraph and sentence markup in document order, so a document opens with its printed masthead. A seeded uniform draw over the documents gives 100 test, 128 validation and $16{,}384$ training documents. Draws shorter than 8k characters are rejected and the draw continues, which rejects $12{,}918$ of the $29{,}530$ draws. The test block is fixed by the seed. The validation and training blocks exist for the in-corpus rows of Table 3 and Table 4. Nothing scored in the pretraining-mixture rows was fit on them. Scoring takes the first 512 tokens of each document.

The corpus was chosen for its provenance. A book corpus would be part of the Pile through Gutenberg, which makes up 9% of the pile mix, and a corpus of scientific papers part of both pretraining mixtures, through ArXiv, another 9% of the pile mix, and through peS2o, 2.3% of Dolma. UNPC is a component of neither: neither corpus lists it, or anything drawn from it, among its sources. Its release channel alone would not rule it out, since the Pile takes two of its components from machine-translation corpus releases, EuroParl and, in the original release, OpenSubtitles from OPUS. UNPC, however, is released as aligned corpus files in XML and TMX format through the UN and OPUS for machine-translation research, and the underlying documents circulate as PDF and Word files on the UN's document system rather than as the HTML pages whose extracted text a web crawl keeps. Overlap with either pretraining corpus is therefore not expected, though we have not verified it document by document, and Common Crawl's coverage of UN web pages is not something we can bound. What holds by construction is narrower: UNPC itself is a component of neither the Pile nor Dolma and was fed to neither fitting mixture, and the scored test block is disjoint from the UNPC validation and training blocks on which the in-corpus rows are fit. The *register* is not new to Pythia: the pile mix carries 1% EuroParl, parliamentary proceedings of a similar institutional kind. The documents themselves are expected to be new to it.

G Experimental details

Algorithm 2 Initialization of the proposal (section 3.7). Pools, the candidate budget N_{cand}, the gauge rows n_g and the sweep cap T are given per protocol in the paragraphs that follow. $\rho = 10^{-3}$.

Require: pooled queries $\{q_i\}$ and keys $\{k_j\}$ of one head, post-RoPE; components C; candidate budget N_{cand}; gauge rows n_g; sweep cap T

1: **Gauge**: $A \leftarrow I$ with the gauge off. With it on, draw n_g query and n_g key rows uniformly; $\Sigma_q \leftarrow S_q + \rho\,(\operatorname{tr} S_q/d)\,I$ with $S_q = \frac{1}{n_g}\sum_i q_i q_i^\top$, and Σ_k likewise; $A \leftarrow M^{1/2}$ for the positive-definite solution M of $M\Sigma_q M = \Sigma_k$, $M = \Sigma_q^{-1/2}\big(\Sigma_q^{1/2}\Sigma_k\Sigma_q^{1/2}\big)^{1/2}\Sigma_q^{-1/2}$, which is symmetric and equalizes the gauged second moments, $A\Sigma_q A = A^{-1}\Sigma_k A^{-1}$. Then $q_i \leftarrow Aq_i$, $k_j \leftarrow A^{-1}k_j$ for every pooled row.

2: $t \leftarrow \lfloor\sqrt{N_{\mathrm{cand}}}\rfloor$; draw t queries and t keys uniformly without replacement, independently

3: candidates $x_{ij} \leftarrow q_i + k_j$ for all t^2 pairs, with weights $w_{ij} \leftarrow e^{q_i^\top k_j} \big/ \sum_{i',j'} e^{q_{i'}^\top k_{j'}}$

4: seed C centres by weighted k-means++: the first drawn $\propto w$, each next $\propto w\,D^2$ with D the distance to the nearest seed

5: **for** up to T sweeps **do**

6: assign each candidate to its nearest centre; **stop** if no assignment changed

7: $\mu_c \leftarrow \sum_{ij\in c} w_{ij}x_{ij} \big/ \sum_{ij\in c} w_{ij}$; a cluster with no mass is re-seeded at the candidate maximizing $w\,D^2$

8: **end for**

9: $\lambda_c \leftarrow \sum_{ij\in c} w_{ij}$, normalized; $\nu_c \leftarrow \log\lambda_c$; $\sigma_c \leftarrow 1$

10: **return** $\theta = (\mu, \log\sigma, \nu)$, whose proposal we write r_0, and A

Training. Queries and keys are taken after the rotary position embedding (RoPE), and attention is causal. Proposals are trained on the training set of Appendix F and selected on its validation set, so neither touches the scored contexts. In Algorithm 1, each weight w_s of line 5 is computed against the full mixture, in the log domain (Blanchard et al., 2021). The gauge A enters only the features of line 7, not the draws or the weights, so its gradient is taken at fixed $\omega_{1:m}$. Under $\mathcal{L}_{\mathrm{CE}}$ the gradient of line 9 is equation 50. Under either surrogate the update is an Adam step (Kingma & Ba, 2015), with a gradient clip on the norm taken jointly over θ and A. The m features of line 5, in training and in scoring, allocate the component index by a stratified pass through λ, one draw per stratum, and draw each component's noise z as an orthogonal block with $\chi(d)$ row norms, the orthogonal random feature (ORF) ensemble FAVOR+ uses. Each index is still marginally $\mathrm{Cat}(\lambda)$ and each z marginally $\mathcal{N}(0, I_d)$, so a draw at a random position follows r_θ, the average over the m draws stays unbiased and Proposition 1 is unchanged. This holds for the GMProp rows of Table 1, Table 2 and Table 3. The proposals of Table 4 trained on UNPC allocate the index the same way but draw z i.i.d., and the runs of Appendix D draw both i.i.d. as in line 5 of Algorithm 1. The next two paragraphs give the settings of each surrogate.

Log-kernel MSE surrogate. For the GMProp rows of Table 1, each head is fit alone on its captured dataset $\mathcal{D}$ of post-RoPE activations, over the causal pairs. Algorithm 2 initializes $C{=}16$ components from a 262144-candidate block, $t{=}512$ per side and $T{=}25$ sweeps, drawn from 8192 query and 8192 key rows of the training contexts. Each step draws $m{=}128$ features and scores 128 contexts from a pool of 2000 on Pythia-70M and 1024 on Pythia-410M and OLMo-1B, at a constant Adam rate of $2{\times}10^{-2}$ and gradient clip 1, for 2000 steps, and 2500 on Pythia-410M, in float64. The with-gauge row initializes A by the closed-form fit on $n_g{=}512$ rows per side and trains it as a rotation times a symmetric positive-definite matrix, each through its matrix logarithm, at $2{\times}10^{-3}$. The no-gauge row fixes $A = I$ with the same seeds and schedule.

Next-token cross-entropy surrogate. For the GMProp rows of Table 2, every head is swapped into a live forward and trained against the frozen model's next-token cross-entropy. The component means are initialized by Algorithm 2 from a 16384-candidate block, $t{=}128$ per side and at most $T{=}25$ sweeps, over the first 128 training contexts with the attention sink excluded. The gauge is initialized by the closed-form fit of Algorithm 2 on $n_g{=}1024$ rows per side from the first 1024 contexts, then trained as an invertible A in the low-rank parameterization equation 39 at rank 64 on Pythia and 128 on OLMo-1B, the full head dimension, at $3{\times}10^{-4}$. Training uses batch 8, $m{=}256$ features, gradient clip 0.5 on Pythia and 2.0 on OLMo-1B, and learning rate $3{\times}10^{-3}$, on a blockwise schedule that trains the means, then the widths together with the gauge, then the mixing coefficients, then all of them at a reduced rate, with phases of 3000, 3600, 2700 and 2700 steps on Pythia, 12,000 planned in all, and 6000, 7200, 5400 and 5400 steps on OLMo-1B, 24,000 planned in all, cosine decay and patience 6. Beyond the surrogate and the set of heads, these runs of Algorithm 1 differ from those of the $\mathcal{L}_{\log}$ rows in the settings above and the blockwise schedule, which worked better for the cross-entropy than training every block together as the $\mathcal{L}_{\log}$ rows do. The no-gauge row of Table 2 is the same run with $A{=}I$. Seeds and the step schedule are held fixed, so the two rows differ only in the gauge. The trained proposals are scored at $m \in \{16, 256, 4096\}$ over seeds $\{0, 1, 2\}$. r_θ carries no dependence on m and is unbiased at each (Proposition 1), so no cell is a separate fit. They are fit at $m{=}256$ and are not retuned for the other two, which makes those cells conservative rather than tuned. Early stopping fired: the Pythia-70M gauged run kept its best proposals at step 9550 of 12,000, and the OLMo-1B gauged run at step 9300 of 24,000.

Gradient of the cross-entropy surrogate. Fix one attention head and write θ for its parameters, $\omega_{1:m}$ for its draws, $\widehat{K}_{ij} = \frac{1}{m}\sum_s \hat{\kappa}_\theta(q_i, k_j; \omega_s)$ for its estimated kernel entries in a live forward, and $\hat{y}_i = \sum_{j\le i} \widehat{K}_{ij} v_j / \sum_{l\le i} \widehat{K}_{il}$ for its output at position i. For a minibatch of contexts, with $\mathcal{L}_{\mathrm{CE}}$ its mean loss, summing over the positions i of every context,

$$\frac{\mathrm{d}\mathcal{L}_{\mathrm{CE}}}{\mathrm{d}\theta} = \sum_i \sum_{j\le i} \frac{\left(\partial \mathcal{L}_{\mathrm{CE}}/\partial \hat{y}_i\right)^\top (v_j - \hat{y}_i)}{\sum_{l\le i} \widehat{K}_{il}} \cdot \frac{1}{m} \sum_{s=1}^{m} \frac{\mathrm{d}\,\hat{\kappa}_\theta(q_i, k_j; \omega_s)}{\mathrm{d}\theta}. \tag{50}$$

The fraction is $\partial \mathcal{L}_{\mathrm{CE}}/\partial \widehat{K}_{ij}$: the row-normalization Jacobian $(v_j - \hat{y}_i)/\sum_l \widehat{K}_{il}$ times $\partial \mathcal{L}_{\mathrm{CE}}/\partial \hat{y}_i$, the gradient backpropagated to this head's output through every later layer of the frozen model. That backpropagated gradient also carries the paths through later heads, whose queries and keys depend

Table 7: Table 1 in full: layer-0 log-kernel MSE (↓), per-head training at both gauge settings.

	Pythia-70M pile mix			Pythia-410M pile mix			OLMo-1B Dolma v1.5		
Method	16	256	4096	16	256	4096	16	256	4096
GMProp, with gauge	**69.35 ± 53.87**	**23.01 ± 20.60**	**9.140 ± 9.048**	**20.49 ± 13.12**	**5.185 ± 4.018**	**1.676 ± 1.496**	**29.39 ± 11.59**	**8.297 ± 3.795**	**2.965 ± 1.547**
GMProp, no gauge	79.40 ± 61.82	27.93 ± 24.90	11.41 ± 11.52	25.08 ± 15.59	6.482 ± 4.480	2.096 ± 1.630	33.58 ± 12.50	10.17 ± 4.37	3.777 ± 2.015
FAVOR#	217.07 ± 197.32	49.50 ± 43.45	14.07 ± 13.76	51.06 ± 46.84	9.320 ± 7.568	2.268 ± 2.004	54.27 ± 26.20	11.52 ± 5.49	3.464 ± 1.934
SimRF	$2.924\times10^3 \pm 3.924\times10^3$	$1.747\times10^3 \pm 2.459\times10^3$	$1.259\times10^3 \pm 1.940\times10^3$	123.84 ± 139.10	55.15 ± 72.15	22.54 ± 30.71	82.47 ± 29.65	30.19 ± 11.40	11.83 ± 5.85
FAVOR+	$2.913\times10^3 \pm 3.842\times10^3$	$1.762\times10^3 \pm 2.448\times10^3$	$1.261\times10^3 \pm 1.950\times10^3$	127.70 ± 139.32	56.64 ± 73.68	21.79 ± 29.61	83.37 ± 30.75	30.84 ± 12.19	11.89 ± 5.95
Coupled PRFs	$2.903\times10^3 \pm 3.939\times10^3$	$1.697\times10^3 \pm 2.392\times10^3$	$1.304\times10^3 \pm 2.025\times10^3$	127.90 ± 138.23	54.10 ± 72.70	21.91 ± 29.01	83.52 ± 30.37	30.16 ± 11.92	11.85 ± 5.90
FAVOR++	$7.616\times10^3 \pm 1.204\times10^4$	$4.475\times10^3 \pm 6.930\times10^3$	$2.797\times10^3 \pm 4.640\times10^3$	251.19 ± 326.80	101.48 ± 132.36	40.36 ± 58.68	122.46 ± 47.56	40.68 ± 15.17	15.91 ± 6.50

on this head's output, so equation 50 is the total derivative for the composed model. The right factor is the two-route derivative of the one-draw estimator, equation 17, unchanged from the log-kernel gradient equation 16, whose outer factor is $2(\log \hat{\kappa}_\theta - q^\top k)/\hat{\kappa}_\theta$ instead.

Hyperparameters. The head dimension is d=64 on Pythia-70M and Pythia-410M and d=128 on OLMo-1B. GMProp has C=16 means on every model. The with-gauge rows initialize A by Algorithm 2 and learn it, the $\mathcal{L}_{\mathrm{CE}}$ rows in the low-rank parameterization at rank d at 3×10^{-4} and the $\mathcal{L}_{\log}$ rows in the dense invertible form at 2×10^{-3}, and the no-gauge rows fix $A = I$. The $\mathcal{L}_{\mathrm{CE}}$ rows train on the blockwise schedule given above for that surrogate, with the best-so-far parameters carried across phase boundaries, and the $\mathcal{L}_{\log}$ rows train all blocks together. Estimator arithmetic is float64 with the model in float32.

Scoring. Scoring reuses the same harness as FAVOR+, pointing at the learned checkpoint and iterating the feature grid over the reported seeds. Table 1 and Table 3 score the kernel estimator on layer 0 only, the one layer whose pair sums are not themselves the output of swapped attention, so every row is scored on the same recorded layer-0 activations, with the log-kernel MSE of Appendix E. Layer-0 cells are mean $\pm$ std over 5 feature-draw seeds pooled over the layer-0 heads. CE cells are the mean over 3 feature-draw seeds, and exact attention, which draws no features, has a single value. CE is computed in one causal pass over each context, as in prefill. With the draws fixed, the causal prefix sums give each position the output that token-by-token decoding would give it, so the CE of GMProp, FAVOR+, SimRF and Coupled PRFs also holds for generation. FAVOR++ and FAVOR# instead solve their closed form once per pass from the queries and keys of every position of every context in it, later positions included, which token-by-token decoding would not yet have.

Baseline configuration. Every baseline is run as published, including any adaptation it performs inside the forward pass. FAVOR# is run as the symmetric dense-exponential map its paper calls SDERF, Theorem 4.3 of (Likhosherstov et al., 2023), and FAVOR++ as the scalar map of the same family, called OPRF in (Likhosherstov et al., 2022). Both re-solve the closed form inside every forward pass from the live queries and keys rather than once from pooled training activations. That is what the online table of Appendix H (Table 12) prices, and the per-head `eigh` charged there is specific to the symmetric variant. FAVOR+, SimRF and Coupled PRFs use no data and fit nothing before deployment.

Table 8: Table 2 in full: next-token CE (↓) and perplexity (↓) on each model's pretraining mixture, every cell with its perplexity.

	Pythia-70M pile mix			Pythia-410M pile mix			OLMo-1B Dolma v1.5		
Method	16	256	4096	16	256	4096	16	256	4096
Exact attention	3.096 (22.10)			2.336 (10.34)			2.463 (11.74)		
GMProp, with gauge	**5.366 (214.0)**	**5.073 (159.6)**	**4.910 (135.7)**	**5.011 (150.1)**	**4.508 (90.71)**	**4.309 (74.36)**	**5.555 (258.5)**	**5.081 (160.9)**	**4.749 (115.4)**
GMProp, no gauge	5.747 (313.3)	5.347 (210.0)	5.096 (163.3)	5.657 (286.4)	4.903 (134.7)	4.681 (107.9)	5.962 (388.5)	5.380 (216.9)	5.043 (154.9)
FAVOR#	6.775 (875.8)	6.160 (473.4)	5.610 (273.0)	7.928 (2773)	6.730 (837.0)	5.695 (297.5)	8.361 (4277)	7.225 (1374)	6.485 (655.3)
SimRF	8.527 (5050)	7.395 (1628)	7.368 (1585)	7.940 (2806)	7.869 (2615)	7.646 (2092)	8.082 (3236)	7.106 (1219)	6.610 (742.1)
FAVOR+	7.759 (2344)	7.584 (1967)	7.302 (1484)	8.282 (3953)	7.848 (2562)	7.693 (2193)	8.625 (5566)	6.981 (1076)	6.448 (631.6)
Coupled PRFs	8.239 (3784)	7.569 (1937)	7.373 (1593)	8.089 (3259)	7.993 (2960)	7.389 (1618)	8.262 (3874)	7.015 (1114)	6.536 (689.4)
FAVOR++	9.570 (1.43×10^4)	8.573 (5285)	8.031 (3076)	8.909 (7397)	8.860 (7043)	8.392 (4411)	8.555 (5191)	7.216 (1361)	6.804 (901.3)

Table 9: Table 3 in full: layer-0 log-kernel MSE (↓) on the UN Parallel Corpus.

	Pythia-70M			Pythia-410M			OLMo-1B		
Method	16	256	4096	16	256	4096	16	256	4096
GMProp, with gauge	**65.53 ± 53.10**	**21.84 ± 19.90**	**8.834 ± 8.915**	**20.63 ± 13.82**	**5.335 ± 4.153**	**1.735 ± 1.522**	**27.47 ± 9.71**	**7.769 ± 3.130**	**2.719 ± 1.233**
GMProp, no gauge	82.95 ± 64.05	28.27 ± 24.73	11.54 ± 10.87	25.67 ± 15.51	6.914 ± 5.057	2.317 ± 1.919	34.19 ± 10.94	10.30 ± 4.15	3.829 ± 1.924
GMProp, with gauge, pretraining mixture	71.84 ± 55.79	24.10 ± 21.92	9.776 ± 9.811	22.37 ± 14.11	5.810 ± 4.489	1.913 ± 1.692	31.95 ± 10.40	9.271 ± 3.658	3.378 ± 1.525
GMProp, no gauge, pretraining mixture	84.13 ± 65.76	29.92 ± 26.65	12.31 ± 12.33	26.83 ± 16.03	7.188 ± 5.030	2.390 ± 1.879	37.78 ± 12.30	11.43 ± 4.62	4.316 ± 2.225
FAVOR#	249.75 ± 223.16	57.21 ± 48.17	16.52 ± 15.56	61.23 ± 55.35	11.52 ± 9.07	2.866 ± 2.437	62.14 ± 25.25	14.45 ± 5.76	4.559 ± 2.177
SimRF	$2.844\times10^3 \pm 3.956\times10^3$	$1.668\times10^3 \pm 2.449\times10^3$	$1.199\times10^3 \pm 1.928\times10^3$	109.99 ± 117.95	47.05 ± 60.39	18.24 ± 23.13	69.26 ± 21.35	23.45 ± 8.16	8.152 ± 3.824
FAVOR+	$2.826\times10^3 \pm 3.861\times10^3$	$1.684\times10^3 \pm 2.436\times10^3$	$1.201\times10^3 \pm 1.939\times10^3$	114.15 ± 120.46	48.14 ± 61.32	17.56 ± 21.87	69.96 ± 22.41	23.89 ± 9.02	8.174 ± 3.863
Coupled PRFs	$2.810\times10^3 \pm 3.955\times10^3$	$1.624\times10^3 \pm 2.383\times10^3$	$1.244\times10^3 \pm 2.017\times10^3$	114.43 ± 119.30	46.35 ± 59.76	17.46 ± 20.48	70.13 ± 21.98	23.38 ± 8.77	8.206 ± 3.703
FAVOR++	$7.565\times10^3 \pm 1.217\times10^4$	$4.417\times10^3 \pm 6.965\times10^3$	$2.782\times10^3 \pm 4.715\times10^3$	238.49 ± 305.57	95.29 ± 120.46	37.54 ± 52.26	106.61 ± 34.84	34.73 ± 11.86	13.26 ± 5.02

Table 10: Table 4 in full: next-token CE (↓) and perplexity (↓) on the UN Parallel Corpus, every cell with its perplexity.

	Pythia-70M			Pythia-410M			OLMo-1B		
Method	16	256	4096	16	256	4096	16	256	4096
Exact attention		3.329 (27.9)			2.503 (12.22)			2.163 (8.696)	
GMProp, with gauge	**5.102 (164.4)**	**4.823 (124.4)**	**4.746 (115.1)**	**4.557 (95.26)**	**4.029 (56.21)**	**3.881 (48.5)**	**4.536 (93.36)**	**4.187 (65.8)**	**4.008 (55.04)**
GMProp, no gauge	5.502 (245.3)	5.157 (173.7)	4.979 (145.3)	5.470 (237.5)	4.780 (119.1)	4.545 (94.19)	5.873 (355.4)	4.910 (135.6)	4.556 (95.24)
GMProp, with gauge, pretraining mixture	5.377 (216.3)	5.135 (169.8)	4.987 (146.5)	5.087 (161.9)	4.616 (101.1)	4.434 (84.3)	5.697 (297.9)	5.139 (170.5)	4.759 (116.6)
GMProp, no gauge, pretraining mixture	5.671 (290.5)	5.324 (205.3)	5.120 (167.3)	5.688 (295.3)	4.924 (137.6)	4.734 (113.7)	6.117 (453.4)	5.389 (218.9)	5.021 (151.5)
FAVOR#	6.616 (746.7)	6.062 (429.1)	5.616 (274.9)	7.686 (2177)	6.587 (725.8)	5.624 (277)	8.450 (4677)	7.197 (1336)	6.469 (645)
SimRF	8.472 (4778)	7.297 (1476)	7.172 (1302)	7.718 (2249)	7.675 (2154)	7.494 (1797)	8.176 (3556)	7.224 (1373)	6.687 (801.8)
FAVOR+	8.549 (5160)	7.344 (1546)	7.182 (1315)	7.745 (2309)	7.815 (2478)	7.433 (1691)	8.295 (4004)	7.250 (1408)	6.685 (800.7)
Coupled PRFs	8.168 (3525)	7.490 (1790)	7.304 (1487)	7.869 (2616)	7.879 (2641)	7.236 (1389)	8.267 (3895)	7.127 (1245)	6.618 (748.8)
FAVOR++	10.201 (2.69×10^4)	8.627 (5579)	8.149 (3461)	8.803 (6653)	8.791 (6573)	8.233 (3761)	8.639 (5649)	7.359 (1570)	6.820 (915.7)

H COST ACCOUNTING

Accuracy at a given m is half the comparison. The other half is what each mechanism spends to reach it. We separate two costs and never net them against each other. ***Offline*** is the one-shot fit before deployment: FLOPs, wall-clock and peak memory. It is zero for every baseline and, for GMProp, the training run of Algorithm 1. ***Online*** is any rebuild from the current queries and keys inside the forward pass: added FLOPs and latency at matched m, non-zero for FAVOR++ and FAVOR# and zero for GMProp and the couplings, which use no data. Each method is scored in its own published regime, so a per-forward closed-form solve is always counted as an online cost.

H.1 THE COST OF THE GAUGE

The gauge's cost is asymptotically negligible whenever $R \leq m$. By the factorization of Appendix C, per token a deployed head spends $O(md)$ on the two cached projections, $O(dR)$ on equation 40, and $O(md)$ folding the feature vector into the running state. With $R \leq m$ the middle term is dominated by the other two, and the totals are a plain $O(Lmd)$ prefill and an $O(md)$ decode, exactly the FAVOR+ entries of Table 5. Learning A changes neither complexity class. The condition is real rather than vacuous: $R \leq d$ by construction, so it binds only when $m < d$, which on these models is the m=16 column.

In practice the overhead is a few percent, because the gauge's cost does not grow with m. The with-gauge and no-gauge rows of Table 12 and Table 13 are the same proposal, the same draw and the same harness differing only in A, so their difference *is* the gauge. At m=256 and L=2048 the prefill costs 4.1% more attention FLOPs on Pythia-70M, 3.6% on Pythia-410M, for 0.01–0.04s of wall-clock and 0.01–0.04GiB of peak memory. Per generated token it is 0.1–0.3ms and no change at all to the recurring state, which holds $m(d{+}2)$ floats whatever A is. The reason those numbers are small is not tuning. Only one of the four token-side products depends on the gauge at all, and that one is a d-dimensional operation competing against m-dimensional ones: against the $\approx 4md$ multiply–adds a head spends per position, a dense $d \times d$ form is $d/4m$, which at d=64, m=256 is 6.25%, and the factored form equation 40 is $\approx R/2m$, cheaper than dense only for $R < d/2$, and 12.5% at the reported R=d. The measured 4.1% is below both counts. The gauge is a fixed d-sized cost amortized against a budget that grows with m, so its share shrinks in exactly the regime the estimator is run in. The cells above are measured at the rank the reported proposals were trained with, R=64, which on the Pythia models is R=d, the setting where equation 39 costs more than a dense A. Any rank below $d/2$ is cheaper than a dense A in both arithmetic and storage.

H.2 COST OF LEARNING

Table 11 is the *offline* adaptation cost: everything spent once, before deployment, to produce one layer-0 proposal of Table 1 and the checkpoint scored in Table 2 at m=256. Each cell reports FLOPs, wall-clock and peak memory. Step counts are in the caption and in the paragraph below, the artifact size is the $Cd + 2C$ floats of section 3.2, plus d^2 with the gauge, and the per-step context draw is in Appendix G. One proposal costs 5–7$\times 10^{13}$ FLOPs and 4–8 minutes of the same card, two to three orders of magnitude below the full-model fit, and its cost barely changes with the model: it depends on the scoring pool and on d, not on the model.

Table 11: Offline learning cost at m=256: FLOPs / wall-clock / peak memory to produce one layer-0 proposal of Table 1 in the first two rows, trained for 2000 steps, and 2500 on Pythia-410M, without early stopping, and the Table 2 checkpoint in the last two rows, trained with patience 6. Pythia on the pile mix, and OLMo-1B on Dolma v1.5. Wall-clock is the training loop as run, with the float64 estimator on the eager path. The k-means initialization of the first two rows runs on the CPU and adds 1.4–2.2min per head. Peak memory is CUDA allocated. Every cell is measured on one NVIDIA H100 SXM 80GB.

Method	Pythia-70M pile mix	Pythia-410M pile mix	OLMo-1B Dolma v1.5
GMProp, with gauge, one head, log-kernel MSE (Table 1)	6.07×10^{13} / 6.50min / 6.71GiB	7.41×10^{13} / 8.10min / 4.32GiB	7.13×10^{13} / 6.20min / 3.68GiB
GMProp, no gauge, one head, log-kernel MSE (Table 1)	5.17×10^{13} / 4.90min / 6.65GiB	6.31×10^{13} / 5.90min / 4.26GiB	4.71×10^{13} / 4.40min / 3.59GiB
GMProp, with gauge, all heads, next-token CE (Table 2)	7.96×10^{15} / 23.2min / 7.25GiB	6.52×10^{16} / 2.51h / 35.6GiB	1.90×10^{17} / 2.54h / 40.4GiB
GMProp, no gauge, all heads, next-token CE (Table 2)	5.92×10^{15} / 16.3min / 7.03GiB	5.08×10^{16} / 1.85h / 33.9GiB	1.18×10^{17} / 1.48h / 37.8GiB

H.3 Cost of inference: prefill

Table 12 is the *online* cost of one forward pass at m=256, any rebuild from the current queries and keys included. Three choices fix what the cells mean.

- **The attention term only.** The model is bit-identical in every row and carries most of a forward's matmuls, so the cells price only the feature draw, any rebuild and the linear-attention products of equation 3.
- **Exact attention is a reference, not a row.** It runs the model's fused float32 `sdpa` kernel, while every estimator row is the same eager float64 stack, so timing them side by side would compare implementations rather than mechanisms. It enters only through the quantities no kernel changes, its $O(L^2 d)$ attention FLOPs and its $2Ld$-per-head KV cache. For the same reason the FLOP and memory columns carry the argument, and the wall-clock column prices our kernels.
- **The deployed path, estimator in float64.** The cells are taken on the fixed-θ path of Appendix C with the model in float32. The estimator stays in float64 because in float32 FAVOR++ and FAVOR# underflow on whole query rows.

FAVOR+, SimRF and Coupled PRFs spend 1.72×10^{11} attention FLOPs, $2.4\times$ fewer than the 4.12×10^{11} of softmax attention, and the gap grows with L. The three rows agree in every column because SimRF and Coupled PRFs only change how the m draws are correlated, not the work done with them. FAVOR++ and FAVOR# form K, so at L=2048 they come in above the quadratic baseline they were meant to replace, at 1.10×10^{12} and 1.11×10^{12}, and carry 13.04GiB against the deployed rows' 3.39. Refitting the feature map every forward shows up in the wall-clock, not in the FLOPs: FAVOR#'s per-head eigendecomposition adds 0.1% to its FLOPs and 0.21s of wall-clock over FAVOR++, whose map is otherwise identical.

Table 12: Online cost on PG-19 at m=256, L=2048: attention FLOPs / wall-clock / peak memory (GiB) of one forward pass of batch 8×2048 on the deployed path, model float32, estimator float64, on one NVIDIA RTX 5090, wall-clock the median over one batch. Exact attention's attention FLOPs over the same batch are 4.12×10^{11} / 3.30×10^{12} / 4.40×10^{12}. ‡These methods refit their feature map from the current queries and keys on every forward, so nothing can be precomputed, and they have to build the full $L \times L$ matrix $\widehat{K}$, which costs $O(L^2 m)$. §FAVOR++'s feature product underflows whole query rows, so on Pythia-70M 33.3% of its log-kernel calls pay the exact $O(nLm)$ repair. *Measured at batch 4×2048: at 8×2048 these two do not fit the 32GB card, the forward holding 27.7GiB when it requests a further 4GiB $L \times L$ float64 block. Their attention FLOPs are linear in the batch, so at batch 8 they are 7.75×10^{12} (FAVOR#) and 7.46×10^{12} (FAVOR++).

Method	Pythia-70M	Pythia-410M	OLMo-1B
GMProp, with gauge	1.79×10^{11} / 0.260s / 3.40GiB	1.43×10^{12} / 1.94s / 4.83GiB	1.77×10^{12} / 2.42s / 9.07GiB
GMProp, no gauge	1.72×10^{11} / 0.250s / 3.39GiB	1.38×10^{12} / 1.90s / 4.79GiB	1.63×10^{12} / 2.32s / 8.97GiB
FAVOR#‡	1.11×10^{12} / 1.45s / 13.0GiB	8.87×10^{12} / 11.6s / 27.0GiB	3.87×10^{12} / 5.43s / 17.8GiB*
SimRF	1.72×10^{11} / 0.250s / 3.39GiB	1.38×10^{12} / 1.90s / 4.79GiB	1.63×10^{12} / 2.32s / 8.97GiB
FAVOR+	1.72×10^{11} / 0.250s / 3.39GiB	1.38×10^{12} / 1.90s / 4.78GiB	1.63×10^{12} / 2.31s / 8.94GiB
Coupled PRFs	1.72×10^{11} / 0.250s / 3.39GiB	1.38×10^{12} / 1.90s / 4.79GiB	1.63×10^{12} / 2.33s / 8.97GiB
FAVOR++‡§	1.10×10^{12} / 1.24s / 13.0GiB	8.77×10^{12} / 8.67s / 27.0GiB	3.73×10^{12} / 3.68s / 17.8GiB*

H.4 Cost of inference: decode

Because the estimate factorizes as $\phi_Q(q)^\top \phi_K(k)$, the sums over all earlier keys collapse into a running state per head, $S = \sum_{j\le t} \phi_K(k_j) v_j^\top$ and $z = \sum_{j\le t} \phi_K(k_j)$, the prefix-sum recurrence of FAVOR+ (Choromanski et al., 2021). Generating one more token then costs a constant $O(md)$ update of that state, against $O(Ld)$ for softmax attention, whose KV cache grows with every token. Table 13 reports the running state: 6.2MiB at a 2048-token prompt, and the same at a 512-token prompt, since its $m(d+2)$ floats per head, which hold S, z and the per-feature key scale, do not depend on L. A KV cache holds a key and a value of d floats per head for every token seen, so for the same prompts it needs $2Ld$ per head, which is 12.0MiB at 512 tokens and 48.0MiB at 2048 over Pythia-70M's 48 heads: the cache becomes four times larger, while the state that linear attention keeps does not change.

Table 13: Per-generated-token cost on PG-19 at m=256: wall-clock / recurring state / peak memory of one decoded token after a 2048-token prompt, prefill excluded, on one NVIDIA RTX 5090. The recurring state is $m(d{+}2)$ floats per head, holding S, z and the per-feature key scale, and does not grow, where softmax attention keeps a KV cache of $2Ld$ per head, 48.0 / 384 / 512 MiB at this prompt, so the state is nearly $8\times$ smaller, and carrying it in float64 against a float32 cache hides a further factor of two in its favour. Peak memory is dominated by the frozen model and barely separates the rows. ‡No fixed θ, so no incremental step exists: generating a token means re-running the whole prefix, one Table 12 forward per token, which is not comparable to the rows above.

Method	Pythia-70M	Pythia-410M	OLMo-1B
GMProp, with gauge	3.00ms / 6.20MiB / 0.690GiB	10.4ms / 49.5MiB / 2.13GiB	8.50ms / 65.0MiB / 5.17GiB
GMProp, no gauge	2.90ms / 6.20MiB / 0.690GiB	10.1ms / 49.5MiB / 2.10GiB	8.10ms / 65.0MiB / 5.08GiB
FAVOR#‡	–	–	–
SimRF	3.00ms / 6.20MiB / 0.680GiB	10.1ms / 49.5MiB / 2.09GiB	8.30ms / 65.0MiB / 5.08GiB
FAVOR+	3.00ms / 6.20MiB / 0.680GiB	10.0ms / 49.5MiB / 2.08GiB	8.50ms / 65.0MiB / 5.04GiB
Coupled PRFs	3.00ms / 6.20MiB / 0.680GiB	10.0ms / 49.5MiB / 2.09GiB	8.30ms / 65.0MiB / 5.08GiB
FAVOR++‡	–	–	–

H.5 The accuracy-aware cost frontier

The two cost tables are best read as a pair of axes, not as one scatter. Table 11 prices the Table 2 fit on an H100. Table 12 and Table 13 price the deployed path on PG-19 on an RTX 5090. The axes therefore describe different corpora on different hardware, and we do not plot them against each other. The online axis that would belong on such a plot is the recurring state, the KV cache or the running state, rather than wall-clock: as section H.3 records, our wall-clock is set by an eager float64 estimator against a fused float32 kernel and would plot our implementation instead of the mechanism. FLOPs stay in Tables 11–12 as the hardware-independent companion.

The trade-off the paper argues for can still be read off the two axes separately. Every baseline lies on the offline-zero line, so the online axis is the only thing that separates them: FAVOR# pays a per-forward solve and is the sharpest baseline on the kernel and on Pythia, FAVOR++ pays the same solve without any gain over the rows that use no data, and FAVOR+ and the couplings pay nothing there and remain at the untrained isotropic accuracy of Table 2. The two GMProp markers are the only ones off that line, trading a high offline cost for the lowest online one. Whether the trade is worth taking depends on how many tokens are served per fit, which is exactly the quantity a reader can compute from the two axes, and the reason we refuse to collapse them into one number.

I Extended related work

Each lineage of Table 5 is discussed here, together with methods that convert pretrained models and the adaptive importance sampling GMProp specializes, so that section 5 can stay on the three properties it claims.

Notes on Table 5. The prefill of every unbiased row includes a one-time $O(md^2)$ feature draw, by Gram–Schmidt orthogonalization for ORF or by folding GMProp's A, reused thereafter. The extra terms of FAVOR++ and FAVOR# are recomputed from the live queries and keys on every forward. LARA has no causal mode (Zheng et al., 2022), so its entry is EVA's. RA is instead unbiased for the row-normalized output, which no Yes row is. Sampling its per-query mixture needs the exact attention row.

Performer / FAVOR+. Performer estimates the attention kernel with positive or trigonometric random features, so that attention is computed in linear time by the products of equation 3 (Choromanski et al., 2021), building on random Fourier features (Rahimi & Recht, 2007) and reducing variance with orthogonal random features (Yu et al., 2016). This is precisely the estimator we improve: GMProp reduces to FAVOR+ at a neutral initialization (Proposition 2), so it is a strict superset of it. The gap is the isotropic proposal, which the authors compensate by further training the model, which changes the model weights and thereby the computed attention. We close the gap by changing the proposal, not the model.

FAVOR++ and FAVOR#. These keep the model frozen and instead make the feature *map* data-adaptive, solving in closed form for its parameters from the current queries and keys: a scalar under the homogeneity heuristic for FAVOR++ (Likhosherstov et al., 2022), and dense matrix parameters for FAVOR# (Likhosherstov et al., 2023), which contains the former as a special case. Both remain unbiased for $e^{q^\top k}$. FAVOR# is the sharpest baseline in every column of the log-kernel tables and in the Pythia columns of the main-text CE tables, and on OLMo-1B the rows that use no data slightly surpass it in most CE cells. The difference is where the adaptation is spent: they re-solve inside every forward pass, whereas our proposals are fit once offline and rebuild nothing. We run them in their own per-forward regime and price it (Appendix H) rather than holding them to ours. The FAVOR+ row is held to the same standard: its features come from the ORF ensemble that the Performer paper specifies, Gram–Schmidt or QR blocks rescaled to $\chi(d)$ row norms (Yu et al., 2016), rather than from i.i.d. Gaussians. ORF couples the m draws while leaving each marginally $\mathcal{N}(0, I_d)$, so Proposition 2 still holds. The GMProp rows of Table 1, Table 2 and Table 3 draw their per-component noise as orthogonal blocks in the same way (Appendix G). The proposals of Table 4 trained on UNPC and the runs of Appendix D draw it i.i.d.

Couplings: SimRF, Hybrid RFs, Coupled PRFs. A parallel line reduces variance by correlating the ensemble rather than changing the sampling law: simplex-coupled directions (Reid et al., 2023), a hybrid of trigonometric and positive maps (Choromanski et al., 2022), and optimal-transport couplings of the frequency norms (Reid et al., 2025). These are orthogonal to what we do and are in principle composable with our proposals: they modify the joint law of the m draws while keeping each draw marginally $\mathcal{N}(0, I_d)$, whereas we change the marginal itself. SimRF and Coupled PRFs enter as baselines at zero adaptation cost, which makes them the reference point at zero cost. Hybrid RFs qualify as well but are omitted, for the reason given in section 4.

Learned features: KL-Transformer, Random Feature Attention, DARKFormer and Coulomb particles. KL-Transformer is the closest to GMProp in form (Pal Chowdhury et al., 2022). Its GMM-PRF variant draws each feature as a component mean plus a learned diagonal matrix times standard Gaussian noise, the draw of equation 14 with a width per coordinate in place of the single width, and trains the mixture by gradients through the draws. It applies no importance weight, so the kernel is defined by the mixture itself, as the expectation over its draws of $e^{\omega^\top q - \|q\|^2} e^{\omega^\top k - \|k\|^2}$. Under $\mathcal{N}(0, I_d)$ this kernel is the Gaussian kernel $e^{-\|q-k\|^2/2}$ rather than $e^{q^\top k}$, and the network is trained from scratch with it on Long Range Arena and GLUE classification tasks. Training lowers the variance by shrinking the covariances, and every such change also changes the kernel. In the limit of zero covariances the kernel becomes a fixed one with one feature per component, so their variance reaches zero only by giving up the random-feature kernel. The importance weight of GMProp instead keeps the estimate unbiased for $e^{q^\top k}$ at every θ by Proposition 1, so its mixture changes only the variance. Random Feature Attention (Peng et al., 2021) also learns its sampling law, a per-coordinate scale of Gaussian draws trained through the reparameterized draw, again without an importance weight, so learning the scale changes the kernel. Its authors also report that a model trained with softmax attention and evaluated with random-feature attention performs hardly better than a randomly initialized one. Unbiasedness alone does not prevent this failure, since the unbiased FAVOR+ also stays far from exact attention in Table 2, and GMProp narrows that gap by also lowering the variance. DARKFormer starts from the form $\psi^*(\omega) \propto p_I(\omega)\sqrt{B_q(\omega)B_k(\omega)}$ that the variance-optimal proposal takes under uniform weights over all pairs (Appendix D), but restricts the proposal to a zero-mean Gaussian $\mathcal{N}(0, \Sigma)$ with $\Sigma = M^\top M$ learned (Farzam et al., 2026). Two differences are decisive: the resulting estimator is unbiased for a *different*, whitened kernel $e^{q^\top \Sigma k}$, and their setups train W_Q, W_K, W_V alongside Σ. A zero-mean Gaussian is never the optimum: by Theorem 1 ψ^* is the root of a kernel-weighted sum of squared Gaussians at the pair sums, and by Lemma 1 any other density has a strictly larger second moment. The Coulomb particle model instead learns the feature vectors Ω themselves by a kernel-alignment and repulsion objective, fixes them, and trains the rest of the network (Badiei Khuzani et al., 2026). Because the learned object is a finite particle set, 64 and 256 features are separately trained configurations rather than one artifact at two budgets, and the authors describe the result as a task-adaptive kernel rather than an unbiased estimator of softmax.

Converting pretrained models. Scatterbrain (Chen et al., 2021) approximates the attention of pretrained vision models without training and without bias, by adding a locality-sensitive-hashing sparse term to Performer's isotropic random features. Its sampling law is not learned, and its sparse term is rebuilt from each input. LoLCATs (Zhang et al., 2025) first trains linear attentions to match the frozen softmax attentions of a pretrained language model, as GMProp fits its proposals offline, but with deterministic learned feature maps, which do not estimate the kernel without bias, and it then fine-tunes the model with low-rank adapters. GMProp learns only its proposals, keeps the estimate unbiased for $e^{q^\top k}$ and does not update the model's weights.

LARA and EVA. Recasting random-feature attention as self-normalized importance sampling, Zheng et al. observe that softmax attention *as a whole* is an expectation under a *per-query* mixture whose means are the pair sums $q_i + k_j$ weighted by the attention row (Zheng et al., 2022). At one query this is the pair-sum mixture G of Lemma 3. GMProp's means are initialized by clustering the same pair sums. The difference is what the mixture may depend on: theirs is rebuilt from the current tokens and the model is trained with the mechanism in place, whereas ours is a fixed summary of the head's own pair sums on a frozen network. And kernel unbiasedness is dismissed rather than traded: RA samples the per-query law directly and is unbiased for the row-normalized output but quadratic, while LARA's self-normalized combination and EVA's control variates leave the head biased (Zheng et al., 2023). Neither direction therefore asks whether that guarantee survives when only the proposal changes.

Variance-minimizing adaptive importance sampling. For $\mu = \mathbb{E}_{X\sim p}[f(X)]$ and a parametric proposal q_θ, the unbiased estimator $f(X)p(X)/q_\theta(X)$ has per-sample variance $M_2(\theta) - \mu^2$ with $M_2(\theta) = \int f(x)^2 p(x)^2/q_\theta(x)\,\mathrm{d}x$. Since μ does not depend on θ, minimizing the variance *is* minimizing the second moment. Writing $Z = \mathbb{E}_{X\sim p}[|f(X)|]$ and $q^\star(x) = |f(x)|\,p(x)/Z$ turns that into an exact statement about densities, $M_2(q) = Z^2\big[1 + \chi^2(q^\star\|q)\big]$, so minimizing M_2 is a forward χ^2-projection of the integrand-aware optimum onto whatever family is being searched, $\arg\min_\theta \chi^2(q^\star\|q_\theta) = \arg\min_\theta \int q^\star(x)^2/q_\theta(x)\,\mathrm{d}x$. Ryu and Boyd optimize exactly this objective by stochastic convex programming, since it is convex in an exponential family's natural parameter, and obtain a proposal that is asymptotically variance-optimal within that family (Ryu & Boyd, 2014). The stochastic-approximation and finite-budget treatments are Lemaire & Pagès (2010) and Kawai (2018), the convex mixture-weight case is He & Owen (2014), Gaussian-mixture proposals adapted in general mixture classes are Cappé et al. (2008), Owen (2013) covers mixture proposals more broadly, and the asymptotic and non-asymptotic theory for the adaptive version is Delyon & Portier (2018) and Akyildiz & Míguez (2021). On the learned side, Müller et al. (2019) train invertible neural proposals on the same χ^2 criterion for Monte Carlo integration, and Goliński et al. (2019) amortize the proposal across a distribution of integration problems rather than solving one. For random Fourier features of shift-invariant kernels, Li et al. (2021) draw the frequencies from a data-dependent proposal with importance weights, which keeps the kernel fixed, and choose the proposal for the risk of kernel ridge regression rather than for a pointwise estimate of $e^{q^\top k}$. GMProp is variance-minimizing adaptive importance sampling specialized to the attention kernel: $q^\star$ is the ψ^* of equation 20, Lemma 1 is the χ^2 identity, and Theorem 2 is the statement that the family can approach the projection. We do not, however, *train* on that objective: section 3.4 minimizes log-kernel MSE or next-token cross-entropy, each with a reference the optimizer cannot change, and Appendix D reports why the direct route fails on real activations.

Two features of that literature matter for what we do with it. First, the gradient may be taken as the score-function estimator $\nabla_\theta M_2 = -\mathbb{E}_{X\sim q_\theta}\big[\big(f(X)p(X)/q_\theta(X)\big)^2 \nabla_\theta \log q_\theta(X)\big]$, the REINFORCE estimator of Williams (1992), which needs only that q_θ be evaluable and is the route Ryu & Boyd (2014) take, but whose variance is its known weakness. Or it may be taken pathwise when q_θ admits a sampling path whose noise does not depend on θ (Mohamed et al., 2020; Rezende et al., 2014), the lower-variance estimator of the two (Xu et al., 2019). Ours is the latter (equation 17): a Gaussian mixture admits it because, given its component index, a draw is standard Gaussian noise, which does not depend on θ, shifted and scaled by θ. Second, the objective contains q_θ^{-1} while the second moment of its gradient contains q_θ^{-3}, so a proposal can give a usable importance sampler and an unusable optimization signal. That asymmetry is why defensive mixtures and heavy-tailed safety components are design requirements in this line rather than implementation details (Hesterberg,

1995; Delyon & Portier, 2021; Korba & Portier, 2022), and why weight diagnostics are reported next to the estimate (Vehtari et al., 2024).