A Coulomb Particle Model for Learning Kernel Attention in Transformers

Masoud Badiei Khuzani MBADIEIKHUZANI@EBAY.COM
eBay Inc., San Jose, CA, USA

Sharath Honnaiah SHONNAIAH@EBAY.COM
eBay Inc., San Jose, CA, USA

Atiq Islam ATISLAM@EBAY.COM
eBay Inc., San Jose, CA, USA

Alex Cozzi ACOZZI@EBAY.COM
eBay Inc., San Jose, CA, USA

Abraham Bagherjeiran ABAGHERJEIRAN@EBAY.COM
eBay Inc., San Jose, CA, USA

Abstract

Randomized features provide a scalable approximation to kernel machines, but their performance depends strongly on the choice of feature distribution. We propose a particle-based method that learns this distribution by optimizing kernel-target alignment while regularizing particles with a Riesz/Coulomb repulsive potential. The resulting Hamiltonian yields diverse, task-adaptive random features and admits a mean-field description through a McKean–Vlasov equation. We instantiate the method in linearized Transformer attention by learning positive random-feature maps in a first alignment phase, then freezing the kernel and training the remaining network parameters with cross-entropy. Experiments on synthetic classification and sentence-level benchmarks show that learned kernelized attention can improve accuracy, calibration, and robustness for several feature maps while preserving linear-attention inference complexity.

1. Introduction

Kernel methods are a principled way to encode nonlinear similarity, but classical training and prediction can scale poorly with the number of samples. Random Fourier features alleviate this problem by replacing an implicit kernel with an explicit randomized map [27]. Nevertheless, the kernel and its associated feature distribution are typically fixed before any labels are observed—a choice that can dominate downstream performance when the appropriate similarity structure is unknown.

We study a supervised kernel-learning procedure that keeps the computational advantages of random features while learning the feature distribution from labels. The key idea is to view the random features as interacting particles. A target-alignment objective attracts particles toward features that explain the labels, while a Riesz/Coulomb repulsive energy prevents collapse and encourages diversity. This gives a concrete optimization algorithm, a statistical-mechanics interpretation, and a direct route to kernelized attention in Transformers.

Our contributions are: (i) a particle Hamiltonian for target-aligned random-feature learning; (ii) a Langevin optimization procedure with a mean-field continuum limit; (iii) an instantiation for positive

© M.B. Khuzani, S. Honnaiah, A. Islam, A. Cozzi & A. Bagherjeiran.

Algorithm 1: Projected Langevin feature learning

Data: particles $\{\boldsymbol{\omega}_k^0\}_{k=1}^N$, step η, inverse temperature β, threshold δ
Result: learned empirical feature law μ_N and kernel $\widehat{K}$
Initialize $\mu_N^0 = N^{-1}\sum_k \delta_{\boldsymbol{\omega}_k^0}$
repeat
  for $k = 1, \dots, N$ **do**
    Draw $\boldsymbol{\xi}_k^m \sim \mathcal{N}(0, I)$ and set $\boldsymbol{\omega}_k^{m+1} = \mathcal{P}_\Omega(\boldsymbol{\omega}_k^m - \eta N \nabla_{\boldsymbol{\omega}_k} \mathcal{H}_N(\boldsymbol{\Omega}_N^m) + \sqrt{2\eta/\beta}\,\boldsymbol{\xi}_k^m)$
  end
  $\mu_N^{m+1} = N^{-1}\sum_k \delta_{\boldsymbol{\omega}_k^{m+1}}$
until $\mathcal{D}(\mu_N^m, \mu_N^{m-1}) < \delta$
Return $\widehat{K}(\boldsymbol{x}, \boldsymbol{x}') = D^{-1}\sum_{k=1}^D \phi_{\boldsymbol{\omega}_k}(\boldsymbol{x})\phi_{\boldsymbol{\omega}_k}(\boldsymbol{x}')$

random-feature attention that preserves linear-time inference; and (iv) empirical evidence that the learned kernels improve several accuracy and calibration metrics on synthetic and NLP benchmarks. Full related work, proofs, and additional ablations are in the supplementary material.

2. Particle kernel learning

Let $\phi : \mathcal{X} \times \Omega \to [-1, 1]$ be a random feature map and let μ be a probability measure over feature parameters. The induced kernel is

$$K_\mu(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{\omega}\sim\mu}[\phi_{\boldsymbol{\omega}}(\boldsymbol{x})\phi_{\boldsymbol{\omega}}(\boldsymbol{x}')]. \tag{2.1}$$

Given training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, we learn μ by maximizing kernel-target alignment [10]. With particles $\boldsymbol{\Omega}_N = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_N)$ and empirical measure $\mu_N = N^{-1}\sum_{k=1}^N \delta_{\boldsymbol{\omega}_k}$, this becomes the finite-dimensional Hamiltonian

$$\mathcal{H}_N(\boldsymbol{\Omega}_N) = -\frac{1}{n(n-1)}\sum_{i\neq j} y_i y_j \frac{1}{N}\sum_{k=1}^N \phi_{\boldsymbol{\omega}_k}(\boldsymbol{x}_i)\phi_{\boldsymbol{\omega}_k}(\boldsymbol{x}_j) + \lambda \mathcal{W}_{N,s}(\boldsymbol{\Omega}_N), \tag{2.2}$$

where

$$\mathcal{W}_{N,s}(\boldsymbol{\Omega}_N) = \frac{1}{2N(N-1)}\sum_{k\neq\ell} g_s(\boldsymbol{\omega}_k - \boldsymbol{\omega}_\ell), \qquad g_s(\boldsymbol{\omega}) = \begin{cases} \|\boldsymbol{\omega}\|_2^{-s}, & s > 0, \\ -\log\|\boldsymbol{\omega}\|_2, & s = 0. \end{cases} \tag{2.3}$$

The first term rewards features whose empirical kernel aligns with labels; the second term spreads particles across Ω. For random Fourier features, this energy reduces to a trigonometric alignment objective involving $\cos(\boldsymbol{X}\boldsymbol{\Omega}_N^\top)$ and $\sin(\boldsymbol{X}\boldsymbol{\Omega}_N^\top)$, which makes the objective differentiable and easy to optimize.

3. Kernelized attention

Self-attention can be written as a normalized kernel smoother [22, 33, 35]:

$$\boldsymbol{a}_i(\boldsymbol{X}) = \frac{\sum_{j=0}^\ell K(\boldsymbol{q}_i, \boldsymbol{k}_j)\boldsymbol{v}_j}{\sum_{j=0}^\ell K(\boldsymbol{q}_i, \boldsymbol{k}_j)}, \qquad K(\boldsymbol{q}, \boldsymbol{k}) = \exp(\boldsymbol{q}^\top \boldsymbol{k}/\sqrt{d}). \tag{3.1}$$

Replacing K by a positive random-feature kernel $K_{\boldsymbol{\Omega}_N}(\boldsymbol{q},\boldsymbol{k}) = \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q})^\top \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k})$ gives the linearized attention estimator

$$\widehat{\boldsymbol{a}}_i(\boldsymbol{X}) = \frac{\boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q}_i)^\top \big(\sum_j \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k}_j)\boldsymbol{v}_j^\top\big)}{\boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q}_i)^\top \big(\sum_j \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k}_j)\big)}. \tag{3.2}$$

This preserves the normalized kernel-smoothing form but reduces the per-head sequence-length dependence from quadratic to linear in ℓ, up to the feature dimension [6, 16, 25]. In Phase A, we learn the feature particles $\boldsymbol{\Omega}_N$ by alignment on sequence representations. In Phase B, we freeze $\boldsymbol{\Omega}_N$ and train the remaining Transformer parameters using cross-entropy.

4. Theoretical results: mean-field limit and large deviation principle

The particle view gives both a continuum training dynamics and an equilibrium concentration result. Define the alignment-induced potential

$$V_{\mathcal{D}}(\boldsymbol{\omega}) = -\frac{1}{n(n-1)} \sum_{i\neq j} y_i y_j \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j), \tag{4.1}$$

and the continuum energy

$$\mathcal{E}_s(\mu) = \int V_{\mathcal{D}}(\boldsymbol{\omega})\, d\mu(\boldsymbol{\omega}) + \frac{\lambda}{2} \iint g_s(\boldsymbol{\omega}-\boldsymbol{\omega}')\, d\mu(\boldsymbol{\omega}) d\mu(\boldsymbol{\omega}'). \tag{4.2}$$

The finite-particle optimizer has the following continuum training law.

Theorem 1 (Projected-particle McKean–Vlasov mean-field limit) *Let $\Omega \subset \mathbb{R}^d$ be compact and convex with C^2 boundary and outward normal $\boldsymbol{n}$. Assume $V_{\mathcal{D}} \in C^2(\overline{\Omega})$, a Lipschitz regularized interaction drift, exchangeable $\mu_0^N \Rightarrow \rho_0 d\boldsymbol{\omega}$, and a vanishing projected-Euler error. If*

$$\boldsymbol{\omega}_k^{m+1} = \mathcal{P}_{\overline{\Omega}}\Big(\boldsymbol{\omega}_k^m - \eta_N N \nabla_{\boldsymbol{\omega}_k} \mathcal{H}_N(\boldsymbol{\Omega}_N^m) + \sqrt{2\eta_N/\beta}\, \boldsymbol{\xi}_k^m\Big), \quad \eta_N \downarrow 0, \tag{4.3}$$

then $\mu_t^N = N^{-1}\sum_k \delta_{\boldsymbol{\omega}_k^{\lfloor t/\eta_N \rfloor}} \Rightarrow \rho_t d\boldsymbol{\omega}$ in probability, uniformly on compact time intervals. With $U_t = V_{\mathcal{D}} + \lambda g_s * \rho_t$,

$$\partial_t \rho_t = \nabla\cdot(\rho_t \nabla U_t) + \beta^{-1}\Delta\rho_t, \tag{4.4}$$

$$\rho_t|_{t=0} = \rho_0, \qquad (\rho_t \nabla U_t + \beta^{-1}\nabla\rho_t)\cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega, \tag{4.5}$$

equivalently $\partial_{\boldsymbol{n}}\rho_t + \beta\rho_t \partial_{\boldsymbol{n}} U_t = 0$. Moreover $\rho_t \geq 0$ and $\int_\Omega \rho_t = 1$.

The proof is provided in the supplementary material. The two parts of U_t have distinct roles. The data potential $V_{\mathcal{D}}$ pulls mass toward features that reduce the empirical objective, while the interaction term spreads mass according to the regularized Riesz geometry and prevents all features from collapsing onto the same locations. The diffusion term $\beta^{-1}\Delta\rho_t$ is the continuum trace of the Langevin noise: at finite temperature it encourages exploration and contributes an entropic regularization; in the zero-temperature limit the equation reduces to the deterministic transport law driven by $-\nabla U_t$. Equivalently,

$$\partial_t \rho_t = \nabla\cdot\left(\rho_t \nabla\left(U_t + \beta^{-1}\log\rho_t\right)\right),$$

so (4.5) is the Wasserstein gradient flow of $\mathcal{F}_\beta(\rho) \stackrel{\text{def}}{=} \mathcal{E}_s(\rho) + \beta^{-1}\operatorname{Ent}(\rho \mid \ell)$. For smooth positive solutions the no-flux condition removes the boundary contribution and gives the dissipation identity

$$\frac{d}{dt}\mathcal{F}_\beta(\rho_t) = -\int_\Omega \rho_t \left|\nabla\left(U_t + \beta^{-1}\log\rho_t\right)\right|^2 d\boldsymbol{\omega} \le 0.$$

This identity is useful conceptually: training decreases the continuum free energy, and the only stationary points are self-consistent Gibbs densities of the form

$$\rho_\infty(\boldsymbol{\omega}) = \frac{1}{Z_\infty}\exp\{-\beta[V_{\mathcal{D}}(\boldsymbol{\omega}) + \lambda(g_s * \rho_\infty)(\boldsymbol{\omega})]\},$$

with the boundary condition inherited from projection.

Theorem 2 (Large deviations for learned feature measures) *Let $\Omega \subset \mathbb{R}^d$ be bounded and let $s > d$. If $\boldsymbol{\Omega}_N \sim \mathbb{P}_{N,\beta_N}$ and $\mu_N = N^{-1}\sum_{k=1}^N \delta_{\boldsymbol{\omega}_k}$, then $\{\mu_N\}$ satisfies an LDP on $\mathcal{P}(\Omega)$. When $\beta_N/N \to 1$, the speed is $r_N = N$ and*

$$\mathcal{J}_s(\mu) = \mathcal{E}_s(\mu) + \operatorname{Ent}(\mu \mid \ell) - \inf_\nu\{\mathcal{E}_s(\nu) + \operatorname{Ent}(\nu \mid \ell)\}, \tag{4.6}$$

where $\operatorname{Ent}(\mu \mid \ell) = \int \log(d\mu/d\ell)d\mu$ for $\mu \ll \ell$. When $\beta_N/N \to \infty$, the speed is $r_N = \beta_N$ and

$$\mathcal{J}_s(\mu) = \mathcal{E}_s(\mu) - \inf_\nu \mathcal{E}_s(\nu). \tag{4.7}$$

For Borel $A \subset \mathcal{P}(\Omega)$, the standard LDP hold with rate $\mathcal{J}_s$ and speed r_N.

The supplement proves this by combining Sanov's theorem with Varadhan's lemma after truncating the singular Riesz interaction. Consequently the learned kernel concentrates around the task-adaptive variational kernel; if the relevant rate has a unique minimizer $\mu_s^\star$, then $K_{\mu_N}(\boldsymbol{x}, \boldsymbol{x}') \to K_{\mu_s^\star}(\boldsymbol{x}, \boldsymbol{x}')$ exponentially at speed N or β_N.

5. Experiments

We evaluate on synthetic classification and SST-2, QQP, and Rotten Tomatoes. Synthetic results show that optimized particles outperform fixed random Fourier features and the importance-sampling baseline under moderate noise. The accuracy gap widens with feature budget: at 64 features per head, optimized particles retain roughly 90% of the performance achieved at 256, while fixed random features degrade more sharply. NLP experiments use a two-layer Transformer encoder with hidden size 128, two heads, 256 random features per head, BERT tokenization, batch size 64, and the two-phase alignment-then-cross-entropy protocol. Training uses AdamW with linear warmup and cosine decay; the alignment phase runs for 20% of total steps before switching to cross-entropy fine-tuning. All results are averaged over three random seeds. We report the full detailed result tables for the three text-classification tasks in the main paper. For all rows in the lower block of each table, the attention projection is constrained to $Q = K = \operatorname{reshape}(x)$, so vanilla and learned-kernel rows are matched feature-map comparisons. The reported metrics include accuracy, micro/macro/weighted F1, ROC-AUC, PR-AUC, MCC, balanced accuracy, LogLoss, Brier score, ECE, and wall-clock training time.

Table 1: SST-2 detailed metrics across attention variants. Bold marks the best value per column; lower is better for LogLoss, Brier, ECE, and train time.

Model	Acc↑	F1μ ↑	F1M↑	F1w↑	ROC-AUC↑	PR-AUC↑	MCC↑	BalAcc↑	LogLoss↓	Brier↓	ECE↓	Train(sec)↓
Learned W_q, W_k baselines												
Vaswani-softmaxattn	0.8050	0.8050	0.8050	0.8051	0.8744	0.8856	0.6100	0.8050	0.5083	0.1498	0.3695	86.8
Vanilla-favor-WqWk	0.7833	0.7833	0.7832	0.7832	0.8587	0.8576	0.5676	0.7837	0.5472	0.1592	0.3584	96.5
Performer	0.7798	0.7798	0.7793	0.7791	0.8618	0.8766	0.5656	0.7810	0.5697	0.1666	0.3804	89.0
Linformer	0.7844	0.7844	0.7843	0.7843	0.8569	0.8673	0.5706	0.7850	0.5922	0.1646	0.3782	82.6
No learned W_q, W_k (Q=K=reshape(x)); feature-map variants												
Vanilla-favor	0.8050	0.8050	0.8050	0.8049	0.8841	0.8928	0.6124	0.8057	**0.4362**	0.1388	0.3420	80.5
Kernel-favor	0.7901	0.7901	0.7900	0.7901	0.8734	0.8834	0.5801	0.7899	0.4477	0.1459	**0.3086**	166.0
Vanilla-elu	0.8028	0.8028	0.8019	0.8017	0.8731	0.8803	0.6150	0.8042	0.4938	0.1505	0.3724	**78.8**
Kernel-elu	0.8108	0.8108	0.8107	0.8108	0.8879	0.8913	0.6216	0.8108	0.4502	0.1392	0.3609	164.3
Vanilla-softplus	0.8108	0.8108	0.8108	0.8108	0.8760	0.8837	0.6225	0.8112	0.4582	0.1429	0.3378	88.5
Kernel-softplus	0.8062	0.8062	0.8058	0.8060	0.8861	0.8864	0.6126	0.8057	0.5122	0.1470	0.3920	173.5
Vanilla-sigmoid2	0.7959	0.7959	0.7957	0.7958	0.8786	0.8881	0.5916	0.7956	0.5025	0.1487	0.3744	82.6
Kernel-sigmoid2	0.7924	0.7924	0.7924	0.7924	0.8785	0.8880	0.5863	0.7929	0.4915	0.1516	0.3790	170.1
Vanilla-softmaxfeat	0.8005	0.8005	0.8003	0.8004	0.8683	0.8728	0.6008	0.8001	0.5163	0.1530	0.3621	89.5
Kernel-softmaxfeat	0.8085	0.8085	0.8085	0.8084	0.8858	0.8982	0.6181	0.8089	0.4381	0.1389	0.3526	171.4
Vanilla-cos2	0.8142	0.8142	0.8142	0.8141	**0.8897**	0.8945	0.6305	0.8148	0.4709	0.1395	0.3866	81.5
Kernel-cos2	0.8119	0.8119	0.8118	0.8119	0.8883	0.8976	0.6237	0.8118	0.4422	**0.1366**	0.3557	165.0
Vanilla-porf-softplus	0.8142	0.8142	0.8142	0.8142	0.8883	**0.8983**	0.6285	0.8143	0.4506	0.1380	0.3569	85.7
Kernel-porf-softplus	**0.8188**	**0.8188**	**0.8188**	**0.8188**	**0.8897**	0.8967	**0.6376**	**0.8189**	0.4561	0.1381	0.3734	162.8

Table 2: QQP detailed metrics across attention variants. Bold marks the best value per column; lower is better for LogLoss, Brier, ECE, and train time.

Model	Acc↑	F1μ ↑	F1M↑	F1w↑	ROC-AUC↑	PR-AUC↑	MCC↑	BalAcc↑	LogLoss↓	Brier↓	ECE↓	Train(sec)↓
Learned W_q, W_k baselines												
Vaswani-softmaxattn	0.8005	0.8005	**0.7949**	0.8038	**0.8959**	**0.8266**	**0.6058**	**0.8133**	**0.4112**	0.1350	0.4314	605.9
Vanilla-favor-WqWk	0.7683	0.7683	0.7346	0.7595	0.8313	0.7619	0.4852	0.7248	0.4863	0.1591	0.4383	**539.7**
Performer	0.7751	0.7751	0.7472	0.7693	0.8402	0.7691	0.5030	0.7388	0.4732	0.1547	0.4360	554.1
Linformer	**0.8077**	**0.8077**	0.7944	**0.8082**	0.8828	0.8157	0.5890	0.7960	0.4153	**0.1337**	0.4541	557.1
No learned W_q, W_k (Q=K=reshape(x)); feature-map variants												
Vanilla-favor	0.7626	0.7626	0.7323	0.7560	0.8168	0.7443	0.4741	0.7242	0.5028	0.1637	0.4182	567.8
Kernel-favor	0.7608	0.7608	0.7319	0.7551	0.8158	0.7438	0.4712	0.7246	0.5016	0.1638	**0.4106**	1197.1
Vanilla-elu	0.7797	0.7797	0.7553	0.7757	0.8458	0.7772	0.5155	0.7485	0.4659	0.1518	0.4386	585.1
Kernel-elu	0.7831	0.7831	0.7648	0.7821	0.8533	0.7863	0.5299	0.7625	0.4598	0.1496	0.4474	1110.3
Vanilla-softplus	0.7786	0.7786	0.7522	0.7735	0.8442	0.7747	0.5116	0.7444	0.4656	0.1522	0.4297	633.3
Kernel-softplus	0.7860	0.7860	0.7651	0.7836	0.8564	0.7891	0.5323	0.7603	0.4578	0.1481	0.4582	1100.7
Vanilla-sigmoid2	0.7772	0.7772	0.7580	0.7760	0.8439	0.7728	0.5165	0.7554	0.4664	0.1529	0.4190	591.9
Kernel-sigmoid2	0.7852	0.7852	0.7666	0.7840	0.8549	0.7885	0.5338	0.7639	0.4537	0.1479	0.4415	1156.0
Vanilla-softmaxfeat	0.7801	0.7801	0.7527	0.7744	0.8471	0.7772	0.5143	0.7441	0.4625	0.1512	0.4365	579.9
Kernel-softmaxfeat	0.7909	0.7909	0.7753	0.7909	0.8679	0.7956	0.5506	0.7753	0.4354	0.1418	0.4371	1072.6
Vanilla-cos2	0.7746	0.7746	0.7533	0.7724	0.8422	0.7715	0.5082	0.7492	0.4688	0.1536	0.4249	650.2
Kernel-cos2	0.7842	0.7842	0.7590	0.7795	0.8525	0.7842	0.5245	0.7511	0.4621	0.1497	0.4526	1254.0
Vanilla-porf_softplus	0.7782	0.7782	0.7531	0.7739	0.8434	0.7751	0.5117	0.7461	0.4685	0.1528	0.4401	588.4
Kernel-porf_softplus	0.7841	0.7841	0.7632	0.7817	0.8536	0.7855	0.5284	0.7585	0.4547	0.1483	0.4419	1110.4

Table 3: Rotten Tomatoes detailed metrics across attention variants. Bold marks the best value per column; lower is better for LogLoss, Brier, ECE, and train time.

Model	Acc↑	F1μ ↑	F1M↑	F1w↑	ROC-AUC↑	PR-AUC↑	MCC↑	BalAcc↑	LogLoss↓	Brier↓	ECE↓	Train(sec)↓
Learned W_q, W_k baselines												
Vaswani-softmaxattn	0.6801	0.6801	0.6793	0.6793	0.7492	0.7522	0.3620	0.6801	0.6396	0.2138	0.2751	12.6
Vanilla-favor-WqWk	0.6764	0.6764	0.6758	0.6758	0.7262	0.7162	0.3539	0.6764	0.7265	0.2327	0.3052	13.0
Performer	0.6642	0.6642	0.6616	0.6616	0.7262	0.7075	0.3333	0.6642	0.7477	0.2347	0.3009	14.4
Linformer	0.6614	0.6614	0.6612	0.6612	0.7133	0.7062	0.3230	0.6614	0.6614	0.2260	0.2509	13.7
No learned W_q, W_k (Q=K=reshape(x)); feature-map variants												
Vanilla-favor	0.7083	0.7083	0.7066	0.7066	0.7732	0.7708	0.4214	0.7083	0.5840	0.1977	0.2229	13.7
Kernel-favor	0.6848	0.6848	0.6848	0.6848	0.7597	0.7479	0.3696	0.6848	0.5823	0.1997	0.2193	26.3
Vanilla-elu	0.6895	0.6895	0.6835	0.6835	**0.7844**	**0.7928**	0.3943	0.6895	0.5809	0.1995	0.2435	12.7
Kernel-elu	0.7148	0.7148	0.7140	0.7140	0.7650	0.7586	0.4320	0.7148	0.5860	0.1991	0.2594	26.1
Vanilla-softplus	0.7036	0.7036	0.7023	0.7023	0.7695	0.7580	0.4107	0.7036	0.5838	0.1977	0.2176	12.6
Kernel-softplus	0.7017	0.7017	0.7017	0.7017	0.7743	0.7677	0.4034	0.7017	0.5766	0.1959	0.2517	23.8
Vanilla-sigmoid2	0.7026	0.7026	0.7022	0.7022	0.7714	0.7675	0.4065	0.7026	0.5758	0.1958	0.2408	12.7
Kernel-sigmoid2	0.6876	0.6876	0.6831	0.6831	0.7562	0.7538	0.3864	0.6876	0.6118	0.2091	0.2733	24.0
Vanilla-softmaxfeat	0.6782	0.6782	0.6733	0.6733	0.7664	0.7703	0.3678	0.6782	0.6065	0.2071	0.2610	12.5
Kernel-softmaxfeat	**0.7167**	**0.7167**	**0.7163**	**0.7163**	0.7779	0.7724	**0.4345**	**0.7167**	**0.5697**	**0.1929**	0.2527	24.5
Vanilla-cos2	0.6876	0.6876	0.6855	0.6855	0.7622	0.7586	0.3803	0.6876	0.5897	0.2015	**0.2154**	**12.4**
Kernel-cos2	0.7092	0.7092	0.7091	0.7091	0.7676	0.7680	0.4185	0.7092	0.5836	0.1982	0.2594	26.7
Vanilla-porf_softplus	0.6839	0.6839	0.6832	0.6832	0.7649	0.7637	0.3692	0.6839	0.5858	0.1996	0.2343	12.6
Kernel-porf_softplus	0.6989	0.6989	0.6986	0.6986	0.7702	0.7700	0.3984	0.6989	0.5750	0.1961	0.2378	24.1

Tables 1, 2, and 3 show that kernel learning is feature-map dependent but often improves both discrimination and proper scoring metrics. On SST-2, Kernel-PORF-softplus gives the highest accuracy and MCC, while Kernel-FAVOR gives the lowest ECE. On QQP, Kernel-softmaxfeat is the strongest $Q = K$ model and reduces LogLoss and Brier score relative to its vanilla counterpart. On Rotten Tomatoes, Kernel-softmaxfeat gives the best accuracy, MCC, LogLoss, and Brier score. The training-time column reflects the one-time alignment phase; after particles are learned, the serving-time attention formula remains linear in sequence length.

Additional experiment details. Inputs use the `bert-base-uncased` WordPiece tokenizer, are padded or truncated to 128 tokens, and use mean pooling over final token representations. The Transformer has two encoder layers, hidden size 128, two attention heads, feed-forward dimension 256, dropout 0.1, and 256 random features per head. Phase A optimizes only the feature particles $\mathbf{\Omega}_N$ using the alignment objective with Langevin noise, log-repulsion, norm projection, and early stopping; Phase B freezes the learned particles and trains the remaining classifier with cross-entropy. Thus the extra cost appears during training, while inference uses the same linear-attention structure as the corresponding vanilla feature map.

6. Conclusion

We introduced a Coulomb/Riesz particle model for supervised random-feature learning and applied it to kernelized Transformer attention. The framework links label alignment, repulsive regularization, Langevin optimization, and equilibrium concentration in a single formulation. Empirically, the method improves several random-feature attention variants and provides a practical way to learn task-adaptive linear-attention kernels.

LLM usage statement. The authors used large language model (LLM) tools for writing assistance and code development. All LLM-assisted content was reviewed, verified, and edited by the authors, who take full responsibility for the correctness, originality, citations, proofs, experiments, figures, and final content of this paper.

References

[1] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6, 2004.

[2] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, 2002. doi: 10.1023/A:1012450327387.

[3] Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. Skyformer: Remodel self-attention with gaussian kernel and nyström method. In *Advances in Neural Information Processing Systems*, 2021.

[4] Krzysztof Choromanski, Mark Rowland, and Adrian Weller. Structured orthogonal random features. In *AISTATS*, 2017.

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Q. Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. URL https://arxiv.org/abs/2009.14794.

[6] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tam'as Sarl'os, Peter Hawkins, Jared Q. Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv:1511.07289*, 2015.

[8] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning non-linear combinations of kernels. *Advances in neural information processing systems*, 22, 2009.

[9] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel algorithms. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 239–246, 2010.

[10] Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaswinder Singh Kandola. On kernel-target alignment. In *Advances in neural information processing systems*, volume 14, pages 367–373, 2002.

[11] Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*, volume 38 of *Stochastic Modelling and Applied Probability*. Springer, Berlin, Heidelberg, 2nd edition edition, 2009. ISBN 978-3-642-03310-0. doi: 10.1007/978-3-642-03311-7.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[13] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *NeurIPS*, 2001.

[14] Richard S. Ellis. *Entropy, Large Deviations, and Statistical Mechanics*. Springer Monographs in Mathematics. Springer, June 2005. doi: 10.1007/0-387-28537-0.

[15] Jaswinder Singh Kandola, John Shawe-Taylor, and Nello Cristianini. Optimizing kernel alignment over combinations of kernels. Technical report, University of Southampton, 2002.

[16] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning (PMLR 119)*, 2020.

[17] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

[18] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[19] Ivano Lauriola and Fabio Aiolli. MKLpy: a python-based framework for multiple kernel learning. *arXiv preprint arXiv:2007.09982*, 2020.

[20] Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, and Tie-Yan Liu. Stable, fast and accurate: Kernelized attention with relative positional encoding. *arXiv:2106.12566*, 2021.

[21] Charles A Micchelli and Massimiliano Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

[22] E. A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 1964.

[23] Stefan Ober, Carl Edward Rasmussen, and Mark Van der Wilk. The promises and pitfalls of deep kernel learning. *Journal of Machine Learning Research*, 22(179):1–65, 2021.

[24] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 115–124. Association for Computational Linguistics, 2005.

[25] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. *arXiv:2103.02143*, 2021.

[26] Ali Rahimi and Ben Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NeurIPS*, 2009.

[27] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, pages 1177–1184, 2007.

[28] Alain Rakotomamonjy, Francis R Bach, St'ephane Canu, and Yves Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[29] Samarth Sinha and John C Duchi. Learning kernels with random features. In *Advances in Neural Information Processing Systems*, pages 1298–1306, 2016.

[30] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, 2013.

[31] Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: A unified understanding of transformer's attention via the lens of kernel. In *EMNLP-IJCNLP*, 2019.

[32] S. R. S. Varadhan. *Large Deviations*, volume 27 of *Courant Lecture Notes*. American Mathematical Society, 2016. ISBN 978-1-4704-2580-1.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[34] Alex Wang, Amanpreet Singh, Julian Michael, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 7th International Conference on Learning Representations (Workshop Track)*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

[35] G. S. Watson. Smooth regression analysis. *Sankhyā, Series A*, 1964.

[36] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

[37] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI Conference on Artificial Intelligence*, 2021.

[38] Felix X. Yu, Ananda Theertha Suresh, Krzysztof Choromanski, Daniel Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *NeurIPS*, 2016.

Supplementary Material

Appendix A. Related work

To discuss related work, we first describe the kernel selection problem in the context of supervised learning problem. Consider a set of n feature vectors and labels $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n, \boldsymbol{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$. We have a loss function $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, where $L(\cdot, y)$ is convex for $y \in \mathcal{Y}$, and a reproducing kernel Hilbert space (RKHS) of functions $\mathcal{F}$ with kernel K. The ℓ_2-regularized optimization problem that underlies the learning task of finding a function $f \in \mathcal{F}$ is as follows

$$\text{Primal: } \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n L(f(\boldsymbol{x}_i), y_i) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2, \quad \text{Dual: } \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^n L^*(\alpha_i, y_i) - \frac{1}{2\lambda} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}, \quad \text{(A.1)}$$

where $\|\cdot\|_{\mathcal{H}}$ is the Hilbert space norm, $\boldsymbol{\alpha} \in \mathbb{R}^n$ are dual variables, $L^*(\alpha, y) = \sup_{z \in \mathbb{R}} \{\alpha z - L(z, y)\}$ is the Fenchel conjugate of the loss function L, and $\boldsymbol{K} \stackrel{\text{def}}{=} [K_{ij}] \in \mathbb{R}^{n \times n}$ with $K_{ij} \stackrel{\text{def}}{=} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the kernel matrix.

While the kernel matrix $\mathbf{K}$ could, in principle, be optimized jointly with the dual variables, much of the literature instead focuses on approaches that decouple kernel learning from the estimation of $f \in \mathcal{F}$. A common strategy is to first construct or adapt the kernel—often by maximizing kernel–target alignment, which quantifies the similarity between the kernel and the target—before solving the regularized risk minimization problem (see, e.g., [9, 10, 15, 18]). This alignment is formulated as the following optimization problem:

$$\max_{K \in \mathcal{K}} \frac{1}{n(n-1)} \sum_{1 \le i \ne j \le n} y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad \text{(A.2)}$$

where $\mathcal{K}$ denotes a predefined class of kernel functions. Several approaches have been proposed in the literature for defining the class of kernels $\mathcal{K}$ in kernel-based learning. In Table 4, we provide a summary of common kernel class choices, along with their computational and memory complexities, as well as the corresponding references. Among the proposed approaches, [29] stands out as a framework that seamlessly integrates with the random feature model, specifically by employing importance sampling of random features within the random feature-based kernel class. In contrast, we propose an alternative approach that yields improved performance. This enhancement is achieved by directly optimizing the distribution of random features within a particle optimization framework, as opposed to relying on importance sampling of random features.

Appendix B. Proposed approach

At a high level, we begin with a feature mapping to represent the kernel. Next, we learn a distribution that aligns this mapping with the labels using the kernel-target alignment (KTA) optimization formulated in Eq. (A.2). From this distribution, we sample random features, which are then used in a standard supervised learning framework.

Specifically, let $\phi : \mathcal{X} \times \Omega \to [-1, 1]$, and μ denotes a probability measure on Ω. We define the kernel

$$K_\mu(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{E}_{\boldsymbol{\omega} \sim \mu}[\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i) \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j)], \quad \text{(B.1)}$$

Table 4: Summary of kernel class $\mathcal{K}$ choices in the literature. *Notes.* n: number of training samples; m: number of base kernels; D: number of random features; d: dimensionality of the input feature vectors; L: number of parameters or layers in the nonlinear transformation.

Kernel Class	**Definition**	**Comp. Complexity**	**Memory Complexity**	**References**
Convex combination of base kernels	$\mathcal{K} = \{K = \sum_{i=1}^m w_i K_i \mid w_i \geq 0,\ \sum_{i=1}^m w_i = 1\}$ $\boldsymbol{w} = (w_1, \ldots, w_m) \in \mathbb{R}^m$ lies on the probability simplex.	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm)$	[1, 18]
Linear combination of base kernels	$\mathcal{K} = \{K = \sum_{i=1}^m w_i K_i \mid \boldsymbol{w} \in \mathbb{R}^m\}$ Allows negative weights; PSD constraints may be required.	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm)$	[8]
Nonlinear kernel combinations	$\mathcal{K} = \{K(\boldsymbol{x}, \boldsymbol{x}') = \sigma\left(\sum_{i=1}^m w_i K_i(\boldsymbol{x}, \boldsymbol{x}') + b\right)\}$ σ is a nonlinearity (e.g., ReLU, sigmoid); $w_i, b \in \mathbb{R}$.	$\mathcal{O}(nmL)$	$\mathcal{O}(nm)$	[23, 36]
Parameterized kernels	$\mathcal{K} = \left\{K_\gamma(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}'\|^2) \,\middle|\, \gamma \in \Gamma\right\}$ $\Gamma \subset \mathbb{R}_+$ is a bounded interval over which γ is optimized.	$\mathcal{O}(n^2 d)$	$\mathcal{O}(n^2)$	[2]
Random feature-based kernels	$\mathcal{K} = \{K_\mu(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{\omega}\sim\mu}[\phi_{\boldsymbol{\omega}}(\boldsymbol{x})\phi_{\boldsymbol{\omega}}(\boldsymbol{x}')] \mid \mu \in \mathcal{M}\}$ $\phi_{\boldsymbol{\omega}}(\boldsymbol{x})$ is a random feature map; $\mathcal{M}$ is a set of distributions over $\boldsymbol{\omega}$.	$\mathcal{O}(nD)$	$\mathcal{O}(nD)$	[29]
SDP-based kernel learning	$\mathcal{K} = \left\{\boldsymbol{K} \in \mathbb{S}_+^n \,\middle|\, \mathrm{tr}(\boldsymbol{K}) \leq c\right\}$ $\mathbb{S}_+^n$ denotes the set of $n \times n$ PSD matrices; $c > 0$ is a trace constraint.	$\mathcal{O}(n^6)$	$\mathcal{O}(n^2)$	[18]
Structured or hierarchical kernels	$\mathcal{K} = \{K = \sum_{i=1}^m w_i K_i \mid \mathbf{w} \in \mathcal{W}_{\text{structured}}\}$ $\mathcal{W}_{\text{structured}}$ encodes priors such as group sparsity or tree-structured dependencies.	$\mathcal{O}(nm \log m)$	$\mathcal{O}(nm)$	[21, 28]
Mean-field kernel approach	$\mathcal{K} = \{K_\mu(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{\omega}\sim\mu}[\phi_{\boldsymbol{\omega}}(\boldsymbol{x})\phi_{\boldsymbol{\omega}}(\boldsymbol{x}')] \mid \mu \in \mathcal{M}\}$ $\phi_{\boldsymbol{\omega}}(\boldsymbol{x})$ is a random feature map; $\mathcal{M}$ is a set of distributions over $\boldsymbol{\omega}$.	$\mathcal{O}(nD^2)$	$\mathcal{O}(nD)$	This work

where we used the shorthand notation $\phi_{\boldsymbol{\omega}}(\cdot) \stackrel{\text{def}}{=} \phi(\cdot; \boldsymbol{\omega})$. We optimize kernel K_μ over all distributions μ in some (large, nonparametric) set $\mathcal{M}$ of possible distributions on random features

$$\sup_{\mu \in \mathcal{M}} \frac{1}{n(n-1)} \sum_{0 \leq i \neq j \leq n} y_i y_j \mathbb{E}_{\boldsymbol{\omega}\sim\mu}[\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j)]. \tag{B.2}$$

We consider independent, identically distributed (i.i.d.) samples or particles $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N \stackrel{\text{i.i.d.}}{\sim} \mu$.[1] Define the configuration of particles $\boldsymbol{\Omega}_N \stackrel{\text{def}}{=} (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N) \in \Omega^N$, and the associated empirical distribution $\mu_N(\boldsymbol{\Omega}_N) = \frac{1}{N}\sum_{k=1}^N \delta_{\boldsymbol{\omega}_k}(\boldsymbol{\omega})$, where $\delta_{\boldsymbol{\omega}_k}(\cdot)$ is Dirac's delta function concentrated at $\boldsymbol{\omega}_k$. We consider the following regularized optimization problem to estimate the expectation term in Eq. (B.2) by optimizing the samples of the distribution, where the expectation is substituted by the Monte Carlo sample average approximation. In particular, we consider the following *Gibbs point process*

1. To distinguish between training samples and random feature samples, we refer to the latter as particles.

with the Hamiltonian

$$\inf_{\mu \in \mathcal{M}_N} \mathcal{H}_N(\boldsymbol{\Omega}_N) \stackrel{\text{def}}{=} \mathcal{E}_N(\boldsymbol{\Omega}_N) + \lambda \mathcal{W}_{N,s}(\boldsymbol{\Omega}_N), \tag{B.3}$$

where $\lambda > 0$ controls the strength of the interaction, the population loss function is approximated as follows:

$$\mathcal{E}_N(\boldsymbol{\Omega}_N) \stackrel{\text{def}}{=} -\frac{1}{n(n-1)} \sum_{0 \le i \ne j \le n} y_i y_j \frac{1}{N} \sum_{k=1}^{N} \phi_{\boldsymbol{\omega}_k}(\boldsymbol{x}_i) \phi_{\boldsymbol{\omega}_k}(\boldsymbol{x}_j). \tag{B.4}$$

Specifically, in the derivation of the empirical loss in Eq. (B.4), we replace the expectation under μ by integration with respect to the empirical measure $\mu_N \stackrel{\text{def}}{=} \frac{1}{N} \sum_{k=1}^{N} \delta_{\boldsymbol{\omega}_k}$:

$$\begin{aligned} K_\mu(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \int_\Omega \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\, \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j)\, \mu(d\boldsymbol{\omega}) \\ &\approx \int_\Omega \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\, \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j)\, \mu_N(d\boldsymbol{\omega}) = K_{\mu_N}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &= \frac{1}{N} \sum_{k=1}^{N} \phi_{\boldsymbol{\omega}_k}(\boldsymbol{x}_i)\, \phi_{\boldsymbol{\omega}_k}(\boldsymbol{x}_j). \end{aligned} \tag{B.5}$$

The following lemma provides the explicit form of this energy function for the random Fourier feature models in [27]:

Lemma 3 (Empirical KTA equals trigonometric energy with random bias) *Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$. Let $\boldsymbol{\Omega}_N \stackrel{\text{def}}{=} (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N) \in \Omega^N$ be the particle configuration and $\boldsymbol{b} \stackrel{\text{def}}{=} (b_1, \ldots, b_N)^\top$ with $b_k \stackrel{i.i.d.}{\sim} \mathrm{Unif}[0, 2\pi]$. Define $\phi_{\boldsymbol{\omega},b}(\boldsymbol{x}) = \sqrt{2}\cos(\boldsymbol{\omega}^\top \boldsymbol{x} + b)$. Consider*

$$\mathcal{E}_N(\boldsymbol{\Omega}_N) \stackrel{\text{def}}{=} -\frac{1}{n(n-1)} \sum_{0 \le i \ne j \le n} y_i y_j \, \frac{1}{N} \sum_{k=1}^{N} \phi_{\boldsymbol{\omega}_k, b_k}(\boldsymbol{x}_i)\, \phi_{\boldsymbol{\omega}_k, b_k}(\boldsymbol{x}_j). \tag{B.6}$$

Let $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ stack the $\boldsymbol{x}_i^\top$, and write

$$\boldsymbol{A} \;=\; \boldsymbol{X}\boldsymbol{\Omega}_N^\top + \mathbf{1}_n \boldsymbol{b}^\top, \qquad \boldsymbol{C} = \cos(\boldsymbol{A}) \in \mathbb{R}^{n \times N}, \quad \boldsymbol{S} = \sin(\boldsymbol{A}) \in \mathbb{R}^{n \times N},$$

where $\mathbf{1}_n$ is the all-ones vector, and $\cos(\cdot)$ and $\sin(\cdot)$ are applied to each element of the matrix. Then, up to an additive constant independent of $(\boldsymbol{\Omega}_N, \boldsymbol{b})$,

$$\mathcal{E}_N(\boldsymbol{\Omega}_N) \;\equiv\; -\frac{2}{n^2} \left\| \boldsymbol{y}^\top \boldsymbol{C} \right\|_2^2, \tag{B.7}$$

and, taking expectation over the random phases $\boldsymbol{b}$,

$$\mathbb{E}_{\boldsymbol{b}}[\mathcal{E}_N(\boldsymbol{\Omega}_N)] \;\equiv\; -\frac{1}{Nn(n-1)} \left(\left\| \boldsymbol{y}^\top \cos(\boldsymbol{X}\boldsymbol{\Omega}_N^\top) \right\|_2^2 + \left\| \boldsymbol{y}^\top \sin(\boldsymbol{X}\boldsymbol{\Omega}_N^\top) \right\|_2^2 \right) \tag{B.8}$$

$$= -\frac{1}{N\, n(n-1)} \sum_{k=1}^{N} \left| \sum_{i=1}^{n} y_i \, e^{\, i\, \boldsymbol{\omega}_k^\top \boldsymbol{x}_i} \right|^2. \tag{B.9}$$

Algorithm 2: Riesz/Coulomb Particles for Kernel Estimation in Kernel Methods

Data: number of particles N; initial particle positions $\{\boldsymbol{\omega}_k^0\}_{k=1}^N$; step size η; inverse temperature β; threshold δ; divergence $\mathcal{D}(\cdot,\cdot)$; number of random feature samples $D < N$

Result: estimated kernel $\widehat{K}(\boldsymbol{x}, \boldsymbol{x}')$

$\mu_N^0 \leftarrow N^{-1}\sum_{k=1}^N \delta_{\boldsymbol{\omega}_k^0}$;

$m \leftarrow 0$;

repeat

  for $k = 1$ **to** N **do**

    sample $\boldsymbol{\xi}_k^m \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$;

    $\nabla\mathcal{H}_N \leftarrow N\nabla_{\boldsymbol{\omega}_k^m}\mathcal{H}_N(\boldsymbol{\Omega}_N^m)$;

    $\boldsymbol{\omega}_k^{m+1} \leftarrow \mathcal{P}_\Omega\Big(\boldsymbol{\omega}_k^m - \eta\nabla\mathcal{H}_N + \sqrt{2\eta/\beta}\,\boldsymbol{\xi}_k^m\Big)$;

  end

  $\mu_N^{m+1} \leftarrow N^{-1}\sum_{k=1}^N \delta_{\boldsymbol{\omega}_k^{m+1}}$;

  $m \leftarrow m+1$;

until $\mathcal{D}(\mu_N^m, \mu_N^{m-1}) < \delta$;

Compute weights $w_k \leftarrow \exp[-\beta\mathcal{H}(\boldsymbol{\omega}_k^m)]/\sum_{j=1}^N \exp[-\beta\mathcal{H}(\boldsymbol{\omega}_j^m)]$;

Sample D particles $\{\boldsymbol{\omega}_k^*\}_{k=1}^D$ according to $\{w_k\}_{k=1}^N$;

$\widehat{K}(\boldsymbol{x}, \boldsymbol{x}') \leftarrow D^{-1}\sum_{k=1}^D \phi_{\boldsymbol{\omega}_k^*}(\boldsymbol{x})\phi_{\boldsymbol{\omega}_k^*}(\boldsymbol{x}')$;

return $\widehat{K}(\boldsymbol{x}, \boldsymbol{x}')$;

Furthermore, the regularization term in Eq. (B.3) captures the interaction energy of every sample $\boldsymbol{\Omega}_k$ with all the other samples $\boldsymbol{\Omega}_\ell$ in an infinite configuration

$$\mathcal{W}_{N,s}(\boldsymbol{\Omega}_N) = \frac{1}{2N(N-1)} \sum_{1\le k\ne \ell\le N} g_s(\boldsymbol{\omega}_k - \boldsymbol{\omega}_\ell), \tag{B.10}$$

where g_s is the homogeneous potential of degree s,

$$g_s(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \begin{cases} \|\boldsymbol{\omega}\|_2^{-s}, & \text{for } s \in (0, +\infty], \\ -\log\|\boldsymbol{\omega}\|_2, & \text{for } s = 0, \\ -\|\boldsymbol{\omega}\|_2^{-s}, & \text{for } s \in (-2, 0). \end{cases}$$

Note that the Hamiltonian in Eq. (B.3) uses the standard mean-field normalization of the pairwise interaction. Under this normalization, the interaction energy is an average over particle pairs, and its large-N limit is the corresponding continuum Riesz energy. Therefore, we do not introduce an additional factor $N^{s/d}$ in either the loss term or the interaction term. The heuristic spacing $N^{-1/d}$ describes nearest-neighbor distances among N well-distributed particles in d dimensions, whereas the distance between a typical pair of independently sampled particles from a fixed limiting distribution remains $O(1)$. Thus, multiplying the Hamiltonian by $N^{s/d}$ would correspond to a different, N-dependent choice of interaction strength rather than to the mean-field scaling considered here. We regard the coordinates in Ω as nondimensionalized; equivalently, any fixed length-scale normalization, such as replacing $g_s(\boldsymbol{\omega})$ by $g_s(\boldsymbol{\omega}/\ell_0)$, can be absorbed into the coupling parameter λ for $s > 0$,

while in the logarithmic case it changes the energy only by an additive constant and hence leaves the induced dynamics unchanged. Viewing the particles as *charged* samples in Ω, the interaction in Eq. (B) corresponds to the *Riesz potential* for general $s > 0$, with the Coulomb/Newtonian case obtained at $s = d - 2$ for $d \geq 3$ and the logarithmic Coulomb kernel at $s = 0$ in $d = 2$. The sign convention is chosen so that the induced pairwise force is repulsive.

These repulsive interactions act as a regularization mechanism, preventing the particles from collapsing into a single point mass (Dirac delta) and promoting their dispersion across the domain Ω. In particular, they encourage the support of the empirical distribution $\widehat{\mu}_N$ to approximate the support of the true underlying distribution $\Omega = \mathrm{supp}(\mu)$, where $\mathrm{supp}(\cdot)$ denotes the support of a distribution.

From a statistical learning theory perspective, such repulsive forces mitigate sample clustering, thereby enhancing the expressiveness of the kernel. By maintaining spatial diversity among the samples, the kernel can better capture the underlying structure of the data. Numerical simulations confirm that this regularization indeed improves both the stability of the particle system and the generalization performance of kernel-based models.

B.1. Langevin dynamics for efficiently solving Eq. (B.3)

We optimize the positions of the samples in Eq. (B.3) using Langevin dynamics. From an optimization perspective, we model the evolution of the particle system over time steps $m = 0, 1, \ldots, T-1$ according to the following Langevin update rule:

$$\boldsymbol{\omega}_k^{m+1} = \mathcal{P}_\Omega\left(\boldsymbol{\omega}_k^m - \eta\, N \nabla_{\boldsymbol{\omega}_k^m} \mathcal{H}_N(\boldsymbol{\Omega}_N^m) + \sqrt{\frac{2\eta}{\beta_N}} \cdot \boldsymbol{\xi}_k^m\right), \quad k = 1, 2, \cdots, N, \tag{B.11}$$

where $\eta \overset{\text{def}}{=} \eta(n, N)$ is the step size that scales with both the number of training samples n and particles N, β_N is the inverse temperature that depends on the number of particles, $\boldsymbol{\xi}_k^m \overset{\text{i.i.d.}}{\sim} \mathsf{N}(\mathbf{0}, \boldsymbol{I}_{d\times d})$ denotes i.i.d. Gaussian noise with zero mean and identity covariance matrix. Moreover, $\mathcal{P}_\Omega(\cdot)$ is the Euclidean projection onto the set Ω. The particles are initially sampled independently from a probability distribution μ_0, i.e., $\boldsymbol{\omega}_1^0, \ldots, \boldsymbol{\omega}_N^0 \overset{\text{i.i.d.}}{\sim} \mu_0$.

This stochastic update rule blends deterministic gradient descent (on the energy landscape defined by the Hamiltonian $\mathcal{H}_N$) with random perturbations, allowing the system to approximate samples from a Gibbs distribution under appropriate conditions. The projection step ensures the dynamics are constrained to a feasible domain, which may encode structural or regularization constraints critical to the optimization problem. In Algorithm 2, we summarize these steps in a kernel learning algorithm.

Using the random feature samples, we construct the randomized feature map

$$\boldsymbol{\phi}_D(\boldsymbol{x}) = \left(\phi_{\boldsymbol{\omega}_1^*}(\boldsymbol{x}), \ldots, \phi_{\boldsymbol{\omega}_D^*}(\boldsymbol{x})\right), \tag{B.12}$$

and define the corresponding RKHS-based function class as

$$\mathcal{F} = \left\{ f : f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{\sqrt{D}} \langle \boldsymbol{\theta}, \boldsymbol{\phi}_D(\boldsymbol{x}) \rangle,\ \boldsymbol{\theta} \in \Theta \subset \mathbb{R}^D \right\}.$$

Under this construction, the primal objective in Eq. (A.1) transforms into the following finite-dimensional optimization problem:

$$\max_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^{n} L\left(\frac{1}{\sqrt{D}} \boldsymbol{\theta}^T \boldsymbol{\phi}_D(\boldsymbol{x}_i), y_i\right) + \frac{\lambda}{2D} \|\boldsymbol{\theta}\|_2^2.$$

Appendix C. Kernelized attention in transformer architecture

Transformers hinge on the self-attention operation [33]. Let the (embedded) input sequence be $\boldsymbol{X} \stackrel{\text{def}}{=} (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\ell-1}) \in \mathbb{R}^{\ell \times d_x}$, where $\boldsymbol{x}_0 \stackrel{\text{def}}{=} \boldsymbol{x}_{\text{CLS}}$ is the CLS token. Queries, keys, and values are produced by linear maps

$$\boldsymbol{Q}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{W}_Q, \qquad \boldsymbol{K}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{W}_K, \qquad \boldsymbol{V}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{W}_V,$$

with $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{d_x \times d}$ and $\boldsymbol{W}_V \in \mathbb{R}^{d_x \times d_v}$. Standard (scaled dot-product) attention is

$$\boldsymbol{A}(\boldsymbol{X}) = \operatorname{softmax}\Big(\tfrac{1}{\sqrt{d}}\, \boldsymbol{Q}(\boldsymbol{X})\boldsymbol{K}(\boldsymbol{X})^\top\Big)\, \boldsymbol{V}(\boldsymbol{X}).$$

Elementwise, $\boldsymbol{A}(\boldsymbol{X}) = (\boldsymbol{a}_i(\boldsymbol{X}))_{i=0}^{\ell}$, where the i-th output $\boldsymbol{a}_i \in \mathbb{R}^{d_v}$ is a *normalized kernel smoother* (*a.k.a.* Nadaraya–Watson estimator)

$$\boldsymbol{a}_i(\boldsymbol{X}) = \frac{\sum_{j=0}^{\ell} K(\boldsymbol{q}_i, \boldsymbol{k}_j)\, \boldsymbol{v}_j}{\sum_{j=0}^{\ell} K(\boldsymbol{q}_i, \boldsymbol{k}_j)}, \qquad K(\boldsymbol{q}, \boldsymbol{k}) = \exp\Big(\tfrac{1}{\sqrt{d}} \boldsymbol{q}^\top \boldsymbol{k}\Big), \tag{C.1}$$

which connects attention to classical nonparametric regression [22, 35] and to kernel interpretations of Transformers [31]. Thus, softmax attention is attention with a specific positive kernel K; the denominator enforces a convex combination and stabilizes the estimator.

Replacing softmax with any positive kernel K yields a family of attentions with controllable inductive bias (locality, smoothness, anisotropy). When K is approximated via a random feature map $\boldsymbol{\phi}_{\boldsymbol{\Omega}_N} : \mathbb{R}^d \to \mathbb{R}^N,\ \boldsymbol{x} \mapsto \boldsymbol{\phi}_{\boldsymbol{\Omega}_N} \stackrel{\text{def}}{=} N^{-1/2}(\phi_{\boldsymbol{\omega}_i}(\boldsymbol{x}))_{i=1}^{N}$ such that $K_{\boldsymbol{\Omega}_N}(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q})^\top \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k})$, we obtain a *linearized* form [6, 16, 25]:

$$\widehat{\boldsymbol{a}}_i(\boldsymbol{X}) = \frac{\boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q}_i)^\top \Big(\sum_{j=1}^{n} \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k}_j)\, \boldsymbol{v}_j^\top\Big)}{\boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q}_i)^\top \Big(\sum_{j=1}^{n} \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k}_j)\Big)} = \frac{\boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q}_i)^\top \boldsymbol{G}}{\boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{q}_i)^\top \boldsymbol{z}}, \tag{C.2}$$

$$\boldsymbol{G} \stackrel{\text{def}}{=} \sum_{j=0}^{\ell} \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k}_j)\, \boldsymbol{v}_j^\top \in \mathbb{R}^{N \times d_v}, \qquad \boldsymbol{z} \stackrel{\text{def}}{=} \sum_{j=0}^{\ell} \boldsymbol{\phi}_{\boldsymbol{\Omega}_N}(\boldsymbol{k}_j) \in \mathbb{R}^N. \tag{C.3}$$

This reduces complexity from $\mathcal{O}(\ell^2 d)$ to $\mathcal{O}(\ell N + N d_v)$ per head, with memory $\mathcal{O}(N d_v)$, while preserving the normalized kernel-smoothing structure; causal/padding masks apply by omitting the masked terms in the sums. Alternative sub-quadratic routes include Nyström approximations and kernelized attention with relative positional encoding [3, 20, 37].

C.1. Alignment for attention kernel.

Since labels depend on the entire input sequence through $\big(\boldsymbol{Q}(\boldsymbol{X}), \boldsymbol{K}(\boldsymbol{X}), \boldsymbol{V}(\boldsymbol{X})\big)$, the alignment problem departs from the usual random feature setup in Eq. (B.2). For a one-layer Transformer, collect the attention outputs into

$$\boldsymbol{\Phi}(\boldsymbol{X}) \stackrel{\text{def}}{=} \big(\widehat{\boldsymbol{a}}_0(\boldsymbol{X}), \ldots, \widehat{\boldsymbol{a}}_\ell(\boldsymbol{X})\big) \in \mathbb{R}^{\ell \times d_v}.$$

A linear classifier can then be applied either

- at token level for token classification problem (e.g., NER): $\hat{y}_i = \boldsymbol{w}^\top \boldsymbol{a}_i(\boldsymbol{X}) + b$, $i = 1, 2, \ldots, \ell$,
- at sequence level via a pooling operator $P : \mathbb{R}^{\ell \times d_v} \to \mathbb{R}^{d_v}$:

$$\hat{y} = \boldsymbol{w}^\top P\big(\boldsymbol{\Phi}(\boldsymbol{X})\big) + b, \qquad P\big(\boldsymbol{\Phi}(\boldsymbol{X})\big) = \sum_{i=0}^{\ell} \pi_i \, \boldsymbol{a}_i(\boldsymbol{X}), \;\; \boldsymbol{\pi} \in \Delta^\ell.$$

  Here Δ^ℓ is the probability simplex, ensuring a convex combination. Examples: [CLS]: $P(\boldsymbol{\Phi}) = \widehat{\boldsymbol{a}}_0(\boldsymbol{X})$ (i.e., $\pi_0 = 1$, others $\pi_k = 0, \forall 0 < k \leq \ell$) where $\boldsymbol{X} \stackrel{\text{def}}{=} (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell)$; mean pooling: $\pi_i = \frac{1}{\ell}$ for all $i = 1, \cdots, \ell$ and $\pi_0 = 0$. (With padding/masks $m_i \in \{0, 1\}$, use $\pi_i = \frac{m_i}{\sum_{j=1}^n m_j}$ for all $i = 1, \cdots, \ell$.)

This mirros the classical random-feature setting—where a random map $\boldsymbol{\phi}(\boldsymbol{x})$ feeds a linear classifier—but here the attention features $\boldsymbol{\Phi}(\boldsymbol{X})$ (and their pooled variant) are functions of the entire sequence $\boldsymbol{X}$. For sequence-level multi-class classification, given training sequences and labels $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^n$ where the labels are $y_i \in \mathcal{Y} = \{1, 2, \cdots, m\}$, and fixed embedding matrices $(\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V)$, we optimize the target-alignment objective

$$\max_{\mu \in \mathcal{M}_N} \; \mathcal{V}_N^{\text{seq}}(\boldsymbol{\Omega}_N) \; = \frac{1}{n(n-1)} \sum_{1 \leq i < j \leq n} \big(\boldsymbol{e}_{y_i}^\top \boldsymbol{e}_{y_j}\big) \, P(\boldsymbol{\Phi}(\boldsymbol{X}_i))^\top P(\boldsymbol{\Phi}(\boldsymbol{X}_j)), \tag{C.4}$$

where $P(\boldsymbol{\Phi}(\boldsymbol{X}_i)) \in \mathbb{R}^{d_v}$ is the pooled representation of sequence $\boldsymbol{X}_i$, and $\{\boldsymbol{e}_k\}_{k=1}^m$ denotes the standard basis of $\mathbb{R}^m$ which hot-encode the label. Note that the dependence on the particle set $\boldsymbol{\Omega}_N$ (equivalently, on μ) enters only through the sequence embeddings $\boldsymbol{\Phi}(\cdot)$ induced by the attention features and the pooling operator P. In particular, $\boldsymbol{\Omega}_N = (\boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_N)$ is implicit in the definition of attention features $\boldsymbol{\Phi}(\boldsymbol{X})$ through each coordinate $\widehat{\boldsymbol{a}}_i(\boldsymbol{X}), i = 0, \cdots, \ell$. Similarly, for token level classification, consider the training dataset $\{(\boldsymbol{x}_{i,1}, y_{i,1}), \cdots, (\boldsymbol{x}_{i,\ell}, y_{i,\ell}))\}_{i=1}^n$. Then, the target alignment problem reads

$$\max_{\mu \in \mathcal{M}_N} \; \mathcal{V}_N^{\text{tok}}(\boldsymbol{\Omega}_N) = \frac{1}{|\mathcal{U}|(|\mathcal{U}|-1)} \sum_{\substack{(i,t) < (j,s) \\ (i,t),(j,s) \in \mathcal{U}}} \big(\boldsymbol{e}_{y_{i,t}}^\top \boldsymbol{e}_{y_{j,s}}\big) \, \widehat{\boldsymbol{a}}_{i,t}(\boldsymbol{X}_i)^\top \widehat{\boldsymbol{a}}_{j,s}(\boldsymbol{X}_j). \tag{C.5}$$

We minimize the *energy* $\mathcal{V}_N^{\text{seq}}(\boldsymbol{\Omega}_N)$ and $\mathcal{V}_N^{\text{tok}}(\boldsymbol{\Omega}_N)$ in conjunction with the Coloumb/Riesz regularizer, for sequence level and token level classification, respectively.

Remark. In standard Transformer blocks, the attention sublayer is followed by a position-wise feed-forward network (FFN), with each sublayer wrapped by residual connections and layer normalization. A canonical FFN acts independently at each position:

$$\text{FFN}(\boldsymbol{a}_i) = \boldsymbol{W}_2 \, \sigma(\boldsymbol{W}_1 \boldsymbol{a}_i(\boldsymbol{X}) + \boldsymbol{b}_1) + \boldsymbol{b}_2, \quad \boldsymbol{W}_1 \in \mathbb{R}^{d_v \times d_{\text{ff}}}, \; \boldsymbol{W}_2 \in \mathbb{R}^{d_{\text{ff}} \times d_v},$$

with nonlinearity σ (e.g., GELU). Using LN for LayerNorm, the residual/normalization update is, schematically, $\tilde{\boldsymbol{a}}_i = \text{LayerNorm}\big(\boldsymbol{a}_i + \text{FFN}(\boldsymbol{a}_i)\big)$. Because the FFN is parameter-shared across positions and applied pointwise, these operations (i) preserve sequence length and token indices; (ii) leave the attention weights and the kernel-smoothing form that produced $\boldsymbol{a}_i(\boldsymbol{X})$ unchanged; and (iii) implement a learned, per-token reparameterization of attention outputs that improves expressivity and optimization stability. We include FFN, residual, and normalization components in our numerical experiments; the linear classifier discussed above is used to isolate the representation induced by attention and to draw a precise parallel with classical random-feature models.

Appendix D. Positive random-feature maps for linearized attention

We use linearized attention in normalized Nadaraya–Watson form (cf. 17, 33), instantiated by *positive* feature maps $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}_+^M$ so that the kernel $K(\boldsymbol{q}, \boldsymbol{k}) \stackrel{\text{def}}{=} \boldsymbol{\phi}(\boldsymbol{q})^\top \boldsymbol{\phi}(\boldsymbol{k})$ is positive semidefinite and the normalization is well-posed. Throughout, we use a *particle* parameterization with columns $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_M$ of $\boldsymbol{\Omega} \in \mathbb{R}^{d \times M}$ learned from data (as opposed to fixed i.i.d. draws 5, 25). Stabilizations used in practice mirror the code: a temperature $\tau > 0$, a small positive floor, and per-token ℓ_1 re-normalization when stated.

Positive exponential random features (FAVOR). The exponential map

$$\boldsymbol{\phi}_{\boldsymbol{\Omega}}^{\text{FAVOR}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{\sqrt{M}} \Big(\exp(\boldsymbol{\omega}_i^\top \boldsymbol{x} - \tfrac{1}{2}\|\boldsymbol{x}\|_2^2) \Big)_{i=1}^M$$

yields $K(\boldsymbol{q}, \boldsymbol{k}) = \frac{1}{M} \sum_{i=1}^M \exp(\boldsymbol{\omega}_i^\top \boldsymbol{q} + \boldsymbol{\omega}_i^\top \boldsymbol{k} - \frac{1}{2}\|\boldsymbol{q}\|^2 - \frac{1}{2}\|\boldsymbol{k}\|^2) \geq 0$. With $\boldsymbol{\omega}_i \sim \mathcal{N}(0, \mathbf{I})$ and $M \to \infty$ this recovers a Monte Carlo approximation to the softmax kernel [5, 25]; we instead *learn* $\boldsymbol{\Omega}$ to obtain a task-adaptive kernel.

Deterministic positive features (ELU$+1$ with floor + ℓ_1 renorm). Following [17], we use

$$\boldsymbol{\phi}_{\boldsymbol{\Omega}}^{\text{elu}+1}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{Z(\boldsymbol{x})} \Big(1 + \text{ELU}\big((\boldsymbol{\omega}_i^\top \boldsymbol{x})/\tau + b_i\big) \Big)_{i=1}^M, \quad Z(\boldsymbol{x}) \propto \sum_i \max\{\text{ELU}(\cdot) + 1,\ \text{floor}\},$$

where ELU is from [7]. We clamp to a small floor and re-normalize per token so that $\sum_i \phi_i(\boldsymbol{x}) = \sqrt{M}$.

Softplus features (floor + ℓ_1 renorm).

$$\boldsymbol{\phi}_{\boldsymbol{\Omega}}^{\text{softplus}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{Z(\boldsymbol{x})} \Big(\text{softplus}\big((\boldsymbol{\omega}_i^\top \boldsymbol{x})/\tau\big) + \text{floor} \Big)_{i=1}^M,$$

again strictly positive, floor-stabilized, and re-normalized; see [13] for softplus.

Squared-sigmoid features (ℓ_1 renorm).

$$\boldsymbol{\phi}_{\boldsymbol{\Omega}}^{\text{sigmoid2}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{Z(\boldsymbol{x})} \Big(\sigma\big((\boldsymbol{\omega}_i^\top \boldsymbol{x})/\tau\big)^2 \Big)_{i=1}^M, \quad \sigma(t) = \tfrac{1}{1+e^{-t}},$$

which are non-negative and re-normalized per token.

Softmax-over-features (per-token). We also consider a *feature-softmax* map

$$\boldsymbol{\phi}_{\boldsymbol{\Omega}}^{\text{softmaxfeat}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sqrt{M}\, \text{softmax}\big((\boldsymbol{\Omega}^\top \boldsymbol{x})/\tau\big),$$

which is strictly positive and sums to $\sqrt{M}$ by construction.

Cosine-squared random features. Motivated by random Fourier features for shift-invariant kernels [26, 27], we use

$$\boldsymbol{\phi}_{\boldsymbol{\Omega},\boldsymbol{b}}^{\cos 2}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{Z(\boldsymbol{x})} \Big(\cos(\boldsymbol{\omega}_i^\top \boldsymbol{x} + b_i)^2 \Big)_{i=1}^M,$$

with fixed phases $\boldsymbol{b}$; non-negativity is immediate, and we apply per-token ℓ_1 renormalization.

PORF-Softplus (orthogonal initialization). To reduce variance and improve conditioning, we initialize $\mathbf{\Omega}$ with *orthogonal random features* blocks [4, 38] and then apply the softplus map above:

$$\boldsymbol{\phi}_{\mathbf{\Omega}}^{\text{porf-softplus}}(\boldsymbol{x}) \;\equiv\; \boldsymbol{\phi}_{\mathbf{\Omega}}^{\text{softplus}}(\boldsymbol{x}), \quad \mathbf{\Omega}^\top \mathbf{\Omega} \approx \mathbf{I}.$$

The parameters remain learnable after orthogonal initialization.

Learning the kernel via alignment/KTA (Phase A). Beyond a vanilla cross-entropy training of the classifier head (Phase B), we *first* adapt $\mathbf{\Omega}$ with a representation-level objective: either a within-class alignment loss (maximizing same-class similarity) or *Kernel Target Alignment* (KTA; 10) using centered Gram matrices K and label kernel Y. This yields task-adaptive positive kernels while preserving linear-time forward/backward passes. We evaluate on SST-2 from GLUE [34] with BERT tokenization [12], matching the experimental setup in our code.

Implementation details. All maps use a temperature τ, small positive floors where applicable, and (except FAVOR and feature-softmax) per-token ℓ_1 re-normalization to keep $\sum_i \phi_i(\boldsymbol{x}) = \sqrt{M}$. Columns of $\mathbf{\Omega}$ are optionally constrained by column-wise ℓ_2 clipping during Phase A to stabilize learning.

Appendix E. Theoretical results

Before presenting our theoretical results, we first outline the key assumptions that underpin our analysis:

Assumption A.1 (Initial distribution of particles). *The particles are initially sampled independently from a probability distribution μ_0 that admits a Lebesgue density $\rho_0(\boldsymbol{\omega}) = \dfrac{\mu_0(\mathrm{d}\mathbf{\Omega})}{\mathrm{d}\boldsymbol{\omega}}$.*

Assumption A.2 (Constant Temperature). *The temperature parameter β_N remains constant and finite throughout the dynamics for a fixed number of particles N.*

Assumption A.3 (Projection space). *The feature domain $\Omega \subset \mathbb{R}^d$ is compact and convex with C^2 boundary. The projected dynamics use non-absorbing, reflective boundary conditions on $\partial\Omega$.*

Assumption A.4 (Bounded Random Feature Embedding). *Let $\Phi(\boldsymbol{x}) \in L^2(\Omega, \mu_0)$ denote the random feature embedding associated with the kernel $K(\boldsymbol{x}, \boldsymbol{x}')$. We assume that*

$$\sup_{x \in \mathcal{X}} \|\Phi(x)\|_{L^2(\Omega,\mu_0)} = \sup_{x \in \mathcal{X}} \sqrt{K(x,x)} \le L_\Phi < \infty. \tag{E.1}$$

E.1. Continuity equations and equilibrium distribution

Theorem 4 (Projected-particle mean-field continuity equation) *Let $\Omega \subset \mathbb{R}^d$ be compact and convex with C^2 boundary and outward unit normal $\boldsymbol{n}$. Assume that $V \in C^2(\overline{\Omega})$, that g_s is smoothly regularized or the particle system remains collision-free, and that $\mu_N^0 \Rightarrow \rho_0(\boldsymbol{\omega})d\boldsymbol{\omega}$. Consider the mean-field-scaled projected Langevin iteration*

$$\boldsymbol{\omega}_k^{m+1} = \mathcal{P}_{\overline{\Omega}}\left(\boldsymbol{\omega}_k^m - \eta_N \left[\nabla V(\boldsymbol{\omega}_k^m) + \frac{\lambda}{N-1}\sum_{\ell \neq k} \nabla g_s(\boldsymbol{\omega}_k^m - \boldsymbol{\omega}_\ell^m)\right] + \sqrt{2\eta_N/\beta}\,\boldsymbol{\xi}_k^m\right), \tag{E.2}$$

where $\boldsymbol{\xi}_k^m \overset{\text{i.i.d.}}{\sim} \mathsf{N}(\mathbf{0}, I_d)$*,* $\eta_N \downarrow 0$*, and the projected-Euler consistency error vanishes as* $N \to \infty$*. Let* $\mu_t^N = N^{-1}\sum_{k=1}^N \delta_{\boldsymbol{\omega}_k^{\lfloor t/\eta_N \rfloor}}$*. Then* $\mu_t^N \Rightarrow \mu_t = \rho_t(\boldsymbol{\omega})d\boldsymbol{\omega}$ *in probability, uniformly for* t *in compact intervals. The limiting density is governed by the McKean–Vlasov equation*

$$\frac{\partial \rho_t(\boldsymbol{\omega})}{\partial t} = \nabla_{\boldsymbol{\omega}} \cdot [\rho_t(\boldsymbol{\omega})\nabla_{\boldsymbol{\omega}} U_t(\boldsymbol{\omega})] + \frac{1}{\beta}\Delta_{\boldsymbol{\omega}}\rho_t(\boldsymbol{\omega}), \qquad (t, \boldsymbol{\omega}) \in (0, T] \times \Omega, \tag{E.3}$$

where

$$U_t(\boldsymbol{\omega}) = V(\boldsymbol{\omega}) + \lambda \int_\Omega g_s(\boldsymbol{\omega} - \boldsymbol{\omega}')\rho_t(\boldsymbol{\omega}')\, d\boldsymbol{\omega}'. \tag{E.4}$$

The PDE is supplemented by the initial datum, the Robin/no-flux boundary condition, and conservation of probability:

$$\rho_t|_{t=0} = \rho_0, \qquad \left(\rho_t \nabla_{\boldsymbol{\omega}} U_t + \beta^{-1}\nabla_{\boldsymbol{\omega}}\rho_t\right) \cdot \boldsymbol{n} = 0 \quad \text{on } (0, T] \times \partial\Omega, \qquad \rho_t \geq 0, \quad \int_\Omega \rho_t(\boldsymbol{\omega})d\boldsymbol{\omega} = 1. \tag{E.5}$$

Equivalently, for smooth solutions, $\partial_{\boldsymbol{n}}\rho_t + \beta\rho_t\partial_{\boldsymbol{n}}U_t = 0$ *on* $\partial\Omega$*. Here* $V(\boldsymbol{\omega}) = -\mathbb{E}[yy'\phi_{\boldsymbol{\omega}}(\boldsymbol{x})\phi_{\boldsymbol{\omega}}(\boldsymbol{x}')]$ *is the external potential induced by the kernel-target alignment loss.*

The boundary condition in (E.5) is the multidimensional analogue of the Robin condition: it says that the probability flux through the boundary is zero. In the zero-temperature limit ($\beta \to \infty$), the evolution reduces to the reflected deterministic continuity equation

$$\frac{\partial \rho_t(\boldsymbol{\omega})}{\partial t} + \nabla_{\boldsymbol{\omega}} \cdot (\rho_t(\boldsymbol{\omega})\boldsymbol{v}_t(\boldsymbol{\omega})) = 0, \qquad (\rho_t \boldsymbol{v}_t) \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega, \tag{E.6}$$

where $\boldsymbol{v}_t(\boldsymbol{\omega}) = -\nabla_{\boldsymbol{\omega}} U_t(\boldsymbol{\omega})$. This PDE formalism captures the macroscopic evolution of particle densities driven by repulsion, external alignment forces, reflection, and thermal fluctuations.

At equilibrium, for any fixed $\beta_N > 0$ and $N > 0$, the particle configuration is distributed according to the *canonical Gibbs measure*:

$$\mathbb{P}_{N,\beta_N}(\mathrm{d}\boldsymbol{\Omega}_N) = \frac{1}{Z_{N,\beta_N}} \exp\left(-\beta_N \mathcal{H}_N(\boldsymbol{\Omega}_N)\right) \mathbf{1}_{\Omega^N}(\boldsymbol{\Omega}_N)\, \mathrm{d}\boldsymbol{\Omega}_N, \tag{E.7}$$

where $\mathbf{1}_{\Omega^N}(\boldsymbol{\Omega}_N)$ is the indicator function of Ω^N, $\mathrm{d}\boldsymbol{\Omega}_N$ is the Lebesgue measure on $(\mathbb{R}^d)^N$, and $Z_{N,\beta}$ is the *partition function*,

$$Z_{N,\beta_N} \overset{\text{def}}{=} \int_{\Omega^N} \exp\left(-\beta_N \mathcal{H}_N(\boldsymbol{\Omega}_N)\right)\, \mathrm{d}\boldsymbol{\Omega}_N, \tag{E.8}$$

which ensures that the Gibbs measure is properly normalized, i.e., $\int_{\Omega^N} \mathbb{P}_{N,\beta}(\boldsymbol{\Omega}_N)\, \mathrm{d}\boldsymbol{\Omega}_N = 1$. Moreover, the *free energy* is defined as

$$F_s(\beta_N, N, \Omega) \overset{\text{def}}{=} -\frac{1}{\beta_N} \log Z_{N,\beta_N}. \tag{E.9}$$

As $\beta_N \to +\infty$ with $N \to \infty$, the Gibbs measure increasingly concentrates around the minimizer (ground state) of the Hamiltonian $\mathcal{H}_N(\boldsymbol{\Omega}_N)$, which corresponds to the solution of optimization problem in Eq. (B.3).

E.2. Thermodynamic limit of the equilibrated state in the short-range case, $s > d$

Since the kernel function is approximated by the empirical measure μ_N through the Monte Carlo approximation in Eq. (B.5), it is essential to analyze the asymptotic behavior of this empirical measure in order to characterize the limiting properties of the kernel approximation itself. In particular, the statistical fluctuations and concentration properties of μ_N directly determine the accuracy and stability of the resulting kernel-based quantities.

In the setting of interest in Algorithm 2, the empirical measure μ_N arises from a system of interacting *charged* particles evolving under Langevin dynamics. Once the dynamics have reached equilibrium (i.e., after a sufficiently large number m of iterations), the distribution of the particle system converges to the canonical Gibbs measure given in Eq. (E.7). This measure describes the statistical equilibrium of the system, incorporating both the deterministic interaction potential and the stochastic perturbations induced by thermal noise. From the perspective of statistical mechanics, such an equilibrium corresponds to a thermodynamically stable macroscopic state, in which relevant observables become stationary in distribution.

The principal objective of this section is to investigate the asymptotic behavior of the random empirical measure μ_N in this equilibrium regime, particularly as the number of particles N tends to infinity. This regime, known as the *thermodynamic limit* ($N \to \infty$), is of fundamental importance in both statistical mechanics and probability theory, as it establishes the connection between microscopic particle interactions and macroscopic statistical laws.

Our main result is the derivation of a *large deviation principle* (LDP) for μ_N, which characterizes the exponential decay of probabilities of rare deviations from the typical equilibrium distribution. The LDP is governed by a *rate function*, which assigns to each admissible probability measure a nonnegative "cost" quantifying the likelihood of its occurrence in the large-N limit. This framework provides a precise quantitative description of the concentration of μ_N around its equilibrium value, as well as the nature of its fluctuations.

For a comprehensive background on the theory of large deviations, we refer the reader to [11] and [32]. For completeness, we recall the formal definition below, which introduces the notion of an LDP and the associated rate function.

Definition 5 (Large Deviation Principle (LDP)) *Let $(\nu_N)_{N\geq 1}$ be a sequence of probability measures on a Polish space Ω equipped with the Borel σ-algebra $\mathcal{B}(\Omega)$. We say that (ν_N) satisfies a Large Deviation Principle (LDP) at speed r_N with rate function $\mathcal{I} : \Omega \to \mathbb{R}_+$ if, for every Borel set $B \subset \mathcal{B}(\Omega)$,*

$$-\inf_{x\in \mathring{B}} \mathcal{I}(x) \leq \liminf_{N\to\infty} \frac{1}{r_N} \log \nu_N(B) \leq \limsup_{N\to\infty} \frac{1}{r_N} \log \nu_N(B) \leq -\inf_{x\in \overline{B}} \mathcal{I}(x),$$

where $\mathring{B}$ and $\overline{B}$ denote the interior and closure of B, respectively. The functional $\mathcal{I}$ is called a good rate function if it is lower semi-continuous and has compact sub-level sets.

Equipped with Definition 5, we are ready to state the following theorem:

Theorem 6 (Large deviations for empirical measures) *Let $\Omega \subset \mathbb{R}^d$ be a bounded open set. For each $N \geq 1$, let $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\} \subset \Omega$ be a random configuration with law $\mathbb{P}_{N,\beta_N}$ as in (E.7), and define the empirical measure*

$$\mu_N \;=\; \frac{1}{N}\sum_{k=1}^{N} \delta_{\boldsymbol{\omega}_k} \;\in\; \mathcal{P}(\Omega).$$

Then $\{\mu_N\}_{N\geq 1}$ satisfies a large deviation principle on $\mathcal{P}(\Omega)$ endowed with the weak topology, with speed r_N and good rate function given as follows.

Define the energy functional

$$\mathcal{E}_s(\mu) \stackrel{\text{def}}{=} \int_\Omega \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\,\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j)\,\mu(\mathrm{d}\boldsymbol{\omega}) \;+\; \frac{\lambda}{2}\int_\Omega\int_\Omega g_s(\boldsymbol{\omega}-\boldsymbol{\omega}')\,\mu(\mathrm{d}\boldsymbol{\omega})\,\mu(\mathrm{d}\boldsymbol{\omega}'),$$

and the (relative) entropy with respect to Lebesgue measure ℓ on Ω,

$$\mathrm{Ent}(\mu \mid \ell) \stackrel{\text{def}}{=} \begin{cases} \displaystyle\int_\Omega \log\Big(\frac{\mathrm{d}\mu}{\mathrm{d}\ell}(\boldsymbol{\omega})\Big)\,\mu(\mathrm{d}\boldsymbol{\omega}) = \int_\Omega \frac{\mathrm{d}\mu}{\mathrm{d}\ell}(\boldsymbol{\omega})\log\Big(\frac{\mathrm{d}\mu}{\mathrm{d}\ell}(\boldsymbol{\omega})\Big)\,\ell(\mathrm{d}\boldsymbol{\omega}), & \mu \ll \ell, \\ +\infty, & \textit{otherwise.} \end{cases}$$

- *If $\beta_N/N \to 1$ as $N\to\infty$, then the LDP holds with speed $r_N = N$ and rate function*

$$\mathcal{J}_s(\mu) = \Big(\mathcal{E}_s(\mu) + \mathrm{Ent}(\mu \mid \ell)\Big) - \inf_{\nu\in\mathcal{P}(\Omega)}\Big(\mathcal{E}_s(\nu) + \mathrm{Ent}(\nu \mid \ell)\Big).$$

- *If $\beta_N/N \to \infty$ as $N\to\infty$, then the LDP holds with speed $r_N = \beta_N$ and rate function*

$$\mathcal{J}_s(\mu) = \mathcal{E}_s(\mu) - \inf_{\nu\in\mathcal{P}(\Omega)} \mathcal{E}_s(\nu).$$

The proof is provided in Appendix A.4. Theorem 6 implies that for any Borel set $A \subset \mathcal{P}(\Omega)$,

$$\begin{aligned} -\inf_{\mu\in A^\circ}\mathcal{J}_s(\mu) \;&\leq\; \liminf_{N\to\infty}\frac{1}{r_N}\log\mathbb{P}_{N,\beta_N}(\mu_N\in A) \\ &\leq\; \limsup_{N\to\infty}\frac{1}{r_N}\log\mathbb{P}_{N,\beta_N}(\mu_N\in A) \;\leq\; -\inf_{\mu\in\overline{A}}\mathcal{J}_s(\mu), \end{aligned}$$

where A° and $\overline{A}$ denote the interior and closure of A, respectively. The theorem states that, under the Gibbs law, the empirical measure μ_N concentrates (in the weak topology) around the minimizers of the relevant variational functional, and the probability of observing a macroscopic deviation decays exponentially fast at speed r_N. In the regime $\beta_N \sim N$, the rate function contains both the interaction energy $\mathcal{E}_s(\mu)$ and the entropy term $\mathrm{Ent}(\mu \mid \ell)$, capturing the competition between energetic preference for structured configurations and entropic preference for spreading mass. In the low–temperature regime $\beta_N/N \to \infty$, the entropy contribution becomes negligible, so the large deviations are governed purely by $\mathcal{E}_s$ and μ_N concentrates on energy minimizers, with fluctuations suppressed on the faster exponential scale set by β_N.

Appendix F. Numerical Experiments

We present simulations on synthetic datasets as well as experiments on NLP datasets. We plan to release the code for all experiments publicly upon publication.

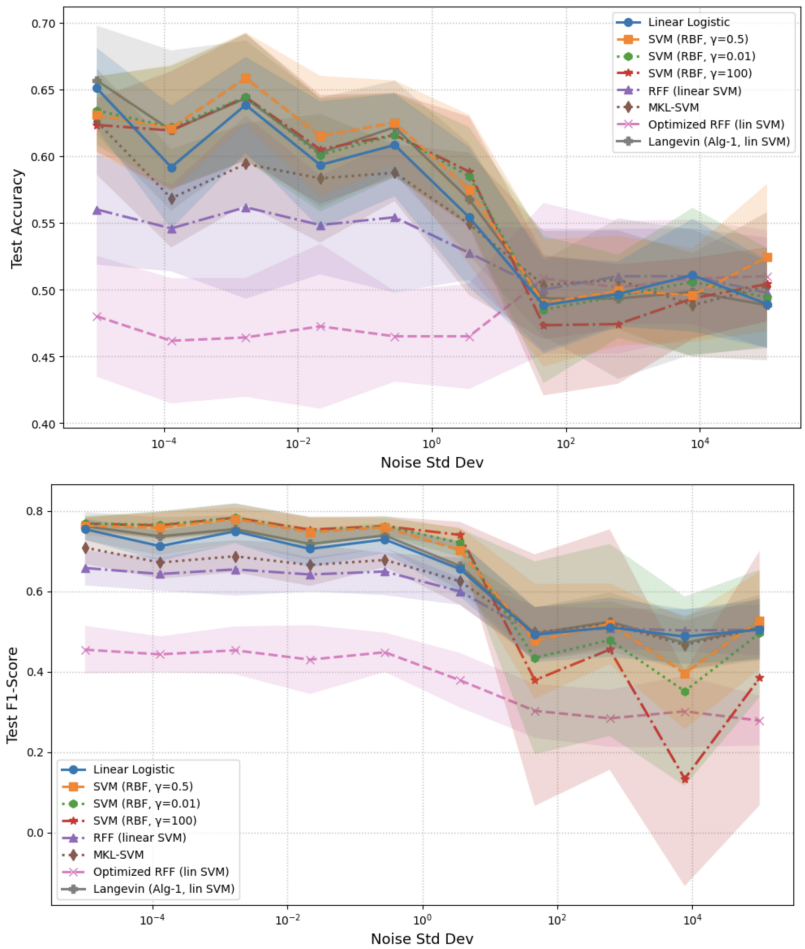

Figure 1: Accuracy (left) and F1 score (right) versus noise standard deviation σ^2 for kernel-learning approaches. Confidence intervals summarize variability across 10 independent trials.

F.1. Nonlinear synthetic classification: data, estimators, and evaluation protocol

F.1.1. PROBLEM SETTING.

We consider binary classification with a fixed design matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ drawn once as $X_{ij} \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0,1)$ under a global seed. Unless stated otherwise we use $n = 400$ and a configurable feature dimension $p \geq 5$. Writing $\boldsymbol{x}_i$ for the i-th row of $\boldsymbol{X}$, the log-odds is

$$\ell(\boldsymbol{x}_i) = 1.5\,\sin(\pi\, x_{i,1}) \;+\; 0.8\, x_{i,2}^2 \;-\; 1.0\, x_{i,3}\, x_{i,4} \;+\; 0.5\,\sin(3\, x_{i,5}) \;+\; \mathbf{1}_{\{p>5\}}\, \boldsymbol{x}_{i,6:p}^{\top} \boldsymbol{w},$$

where the "extra–dimensions" weight vector $\boldsymbol{w} \in \mathbb{R}^{p-5}$ is drawn once from $\mathsf{N}(\mathbf{0}, 0.3^2 \boldsymbol{I})$ (same $\boldsymbol{w}$ for all trials), and $\mathbf{1}_{\{\cdot\}}$ is the indicator function. Labels are generated by corrupting the logit with additive Gaussian noise and passing through a logistic link:

$$\xi_i \sim \mathsf{N}(0, \sigma^2), \qquad y_i \,\big|\, \boldsymbol{x}_i \sim \text{Bernoulli}\big(\varsigma\big(\ell(\boldsymbol{x}_i) + \xi_i\big)\big)\,, \quad \varsigma(t) = \tfrac{1}{1+e^{-t}}.$$

Table 5: Observed wall-clock runtime per job in *seconds*. Values correspond only to measured sample sizes.

Model	**Number of samples** n							
	100	500	1,000	2,000	3,000	4,000	10,000	20,000
Linear Logistic	0.142	0.100	0.0882	0.184	0.173	0.204	0.117	0.121
SVM (RBF kernel)	0.00231	0.0425	0.140	0.0910	0.317	0.468	1.98	9.72
RFF (linear SVM)	0.00332	0.0250	0.0270	0.0326	0.0416	0.0571	0.160	0.335
MKL-SVM	2.06	9.14	29.6	131	482	874	7.21e3	4.49e4
Optimized RFF (lin SVM)	0.0346	0.127	0.299	0.533	0.855	1.04	2.94	4.87
Langevin (Alg-1, lin SVM)	21.6	26.1	45.8	80.5	95.9	124	417	922

Table 6: Power-law fits $t(n) = c\,n^{\gamma}$ summarizing runtime scaling behavior. Exponent γ measures the growth rate, prefactor c sets the baseline runtime, and $\sigma_{\log}$ indicates variability in log-space residuals.

Model	γ	c	$\sigma_{\log}$	n_{obs}
MKL-SVM	1.961	8.00e-05	0.671	8
SVM (RBF kernel)	1.477	3.00e-06	0.503	8
Optimized RFF (lin SVM)	0.967	3.55e-04	0.085	8
RFF (linear SVM)	0.799	9.50e-05	0.319	8
Langevin (Alg-1, lin SVM)	0.730	3.82e-01	0.430	8
Linear Logistic	0.021	1.16e-01	0.290	8

We sweep the *logit noise standard deviation* σ over logarithmically spaced values in $[10^{-5}, 10^{5}]$. A single $70\%/30\%$ train/test split (random state 0) of $\boldsymbol{X}$ is reused across all trials and noise levels; for each σ and trial, only the noise (ξ_i) (and any method-specific randomness) is resampled.

F.1.2. BASELINES AND LEARNED FEATURE MAPS.

All methods use the same train/test partitions of $\boldsymbol{X}$. Random-feature methods are aligned to a common feature budget $D = 200$.

(1) *Linear logistic regression* on the raw inputs $\boldsymbol{X}$ (baseline).

(2) *SVM with Gaussian (RBF) kernel* $K(\boldsymbol{x}, \boldsymbol{x}') = \exp\big(-\gamma\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2\big)$ with fixed $\gamma = 0.5$ and default hinge loss, $C = 1$.

(3) *Random Fourier features (RFF) + linear SVM.* Draw $W_{:,j} \sim \mathsf{N}(0,\ 2\gamma\,\boldsymbol{I}_p)$, $b_j \sim \mathrm{Unif}[0, 2\pi]$, form $z(\boldsymbol{x}) = \sqrt{2/D}\,\cos(W^{\top}\boldsymbol{x} + b)$ with optimized $\gamma = 0.5$ and $D = 200$, then train a linear SVM (hinge loss, $C = 1$).

(4) *Multiple-kernel learning (MKL) + SVM (precomputed).* Build Gaussian Gram matrices with $\gamma \in \{0.1, 0.3, 0.7, 1.0, 1.3, 1.6\}$, apply the MKLPY normalization, learn nonnegative mixture weights via MEMO, and feed the weighted train/test kernels to an SVM with a precomputed kernel.

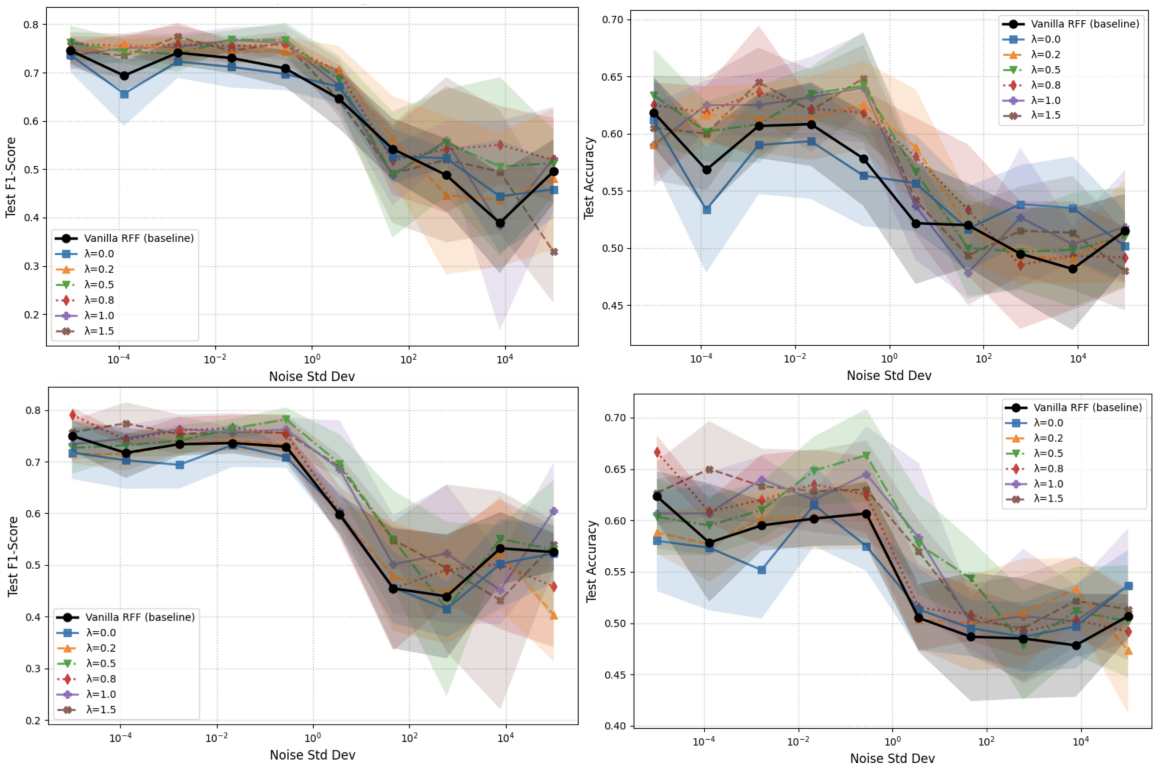

Figure 2: Ablation study of Coulomb (top) and Riesz (bottom) potentials for charged particles. Increasing λ strengthens the repulsive interaction between particles. The baseline uses vanilla RFF with no sampling optimizations. Confidence intervals summarize variability across 10 independent trials.

(5) *Optimized random features (importance sampling) + linear SVM.* Following [29], sample $N_w = 10^4$ candidate features, use divergence threshold $\rho = N_w \cdot 0.005$ and tolerance 10^{-10} to obtain a reweighted set, then align to $D = 200$ via stratified subsampling (padding if needed). Train a linear SVM on the resulting features ($C = 1$).

(6) *Langevin spectral estimators (two variants) + linear SVM.* Evaluate (i) an "Alg-1" Coulomb-gas variant with $\lambda_{\text{reg}} = 0.5$ and (ii) a robust variant with $\lambda_{\text{reg}} = 0$. Both use $N = 300$ particles, feature budget $D = 200$, step size $\eta = 30$, inverse temperature $\beta = 10^2$, horizon $T_{\max} = 2000$, logarithmic repulsion, scaling by N, maximum frequency norm 5.0, and gradient clipping 1.0. After fitting, we form Fourier features from the learned frequencies/phases and train a linear SVM ($C = 1$).

F.1.3. IMPLEMENTATION.

We rely on MKLPY [19] for MKL (mixture-of-RBFs) and on scikit-learn for linear/logistic models and SVMs. The importance-sampling optimized RFF pipeline of [29] is reimplemented in Python. The Langevin estimators are also implemented in Python. We run the simulations on CPU.

F.1.4. TRAINING PROTOCOL, METRICS, AND PLOTS.

For each noise level σ and each of three independent trials, we fit every available method on the training split and evaluate on the test split. We report, for each method and σ, the mean and standard deviation (over trials) of:

- *Test accuracy*: $\mathrm{Acc} = \frac{1}{n_{\mathrm{test}}} \sum_{i \in \mathcal{D}_{\mathrm{test}}} \mathbf{1}\{\hat{y}_i = y_i\}$.
- *Test F1-score*: (binary, with $\mathrm{zero_division} = 0$ in implementation).
- *Wall-clock time per job*: (seconds), measured with a timer from the start of a method's pipeline (feature construction included for RFF variants) to prediction on the test set.

We produce three summary plots versus σ (logarithmic x-axis): (A) accuracy, (B) F1-score, and (C) runtime; shaded bands depict ± 1 standard deviation.

F.1.5. REPRODUCIBILITY AND COMPUTATION.

A global seed (42) fixes $\boldsymbol{X}$, the train/test split, and (when $p > 5$) the extra-dimension weights $\boldsymbol{w}$; per-job seeds are derived deterministically from the noise and trial indices. We cap BLAS threads to one in parent and worker contexts to avoid oversubscription. Jobs indexed by (σ, trial) are executed in parallel using a process pool with up to eight workers; progress is tracked asynchronously. Across methods using random features, the feature budget is held fixed at $D = 200$ to ensure comparable capacity.

F.1.6. RESULTS.

In Figure 1, we report accuracy and F1 score as functions of the additive-noise standard deviation σ. Across noise levels, the SVM with a Gaussian (RBF) kernel—when its bandwidth γ is properly tuned—achieves the best performance. This is expected because the random-feature model only approximates the kernel; with a small number of features ($D = 200$), approximation error degrades classification performance. The same figure also shows that the RBF SVM can underperform substantially when the bandwidth γ is mis-specified, e.g., $\gamma = 0.01$ and $\gamma = 100$ corresponding to green and red dashed lines, respectively.

Moreover, relative to a linear SVM trained on fixed random Fourier features (RFF), jointly optimizing the random features yields clear gains in both accuracy and F1 (purple dashed vs. grey solid curves), confirming the benefit of random-feature optimization. Moreover, compared to alternative importance sampling of random features, we obtain a clear gain by optimization of random feature samples directly (brown line vs purple line).

F.1.7. ABLATION: COULOMB/RIESZ INTERACTION STRENGTH.

We vary the potential strength $\lambda \in \{0.0, 0.2, 0.5, 0.8, 1.0, 1.5\}$ and evaluate robustness as input noise increases (x–axis: noise standard deviation on a log scale). Across both metrics (Accuracy and F1), small noise regimes ($\sigma \leq 10^{-3}$) show minimal separation between curves, indicating that the potential has little effect when the signal is clean. As noise grows to the moderate range ($10^0 - 10^2$), nonzero potentials consistently outperform the vanilla RFF baseline and $\lambda = 0$, with $\lambda \approx 0.5 - 1.0$ yielding the most reliable gains (typically a few points) and smaller variance, suggesting improved stability. In the extreme-noise regime ($\sigma \geq 10^3$), performance degrades for all settings and the gaps

Table 7: Statistics of sentence-level and sentence-pair classification datasets used in our experiments. Counts denote the number of examples per split. For sentence-pair tasks, each example consists of a pair of sentences.

Dataset	Train	Dev	Test	Classes	Task
SST-2 [30]	67,349	872	1,821	2	Sentiment classification
QQP [34]	363,846	40,430	390,965	2	Duplicate question detection
Rotten Tomatoes [24]	8,530	1,066	1,066	2	Sentiment classification

narrow; very large strengths ($\lambda = 1.5$) can oversmooth and underperform, while small-to-moderate λ remains competitive but offers diminishing returns. Overall, a *moderate* Coulomb/Riesz potential ($\lambda \approx 0.5 - 1.0$) provides the best robustness–accuracy trade-off, whereas too weak or too strong potentials are less effective.

F.2. Sentence-Level Classification Benchmarks

Dataset selection. We evaluate sentence-level and sentence-pair classification performance on a set of widely used English benchmarks that are standard in prior work on attention mechanisms.

The selected datasets span multiple semantic phenomena and dataset scales, including binary sentiment analysis on *SST-2* and *Rotten Tomatoes*, as well as semantic equivalence detection on the *QQP* benchmark. Together, these tasks cover both single-sentence and sentence-pair settings, enabling evaluation across varying supervision regimes and levels of linguistic complexity.

Datasets and splits. *SST-2* [30] is a binary sentiment classification task derived from the Stanford Sentiment Treebank. We use the GLUE version of the dataset, which removes neutral examples and provides sentence-level annotations, and follow the standard GLUE train/dev/test splits.

QQP [34] (Quora Question Pairs) is a sentence-pair classification task in which each example consists of two questions from Quora annotated for semantic equivalence. We adopt the official GLUE splits and process each question pair by concatenation with a special separator token.

Rotten Tomatoes [24] is a sentence-level binary sentiment classification dataset constructed from movie reviews. We use the standard polarity version of the dataset and follow the commonly used train/dev/test splits.

Preprocessing. We apply minimal preprocessing to preserve the original linguistic structure of each dataset. All text is tokenized using a fixed WordPiece vocabulary, with original casing and punctuation retained. Inputs are truncated or padded to a maximum length of 128 tokens. For sentence-pair inputs in *QQP*, the two questions are concatenated using a special separator token. No segment embeddings or task-specific features are used unless otherwise stated.

Training protocol. For each dataset, we train models on the official training split and use the validation split exclusively for model selection and early stopping. Hyperparameters are fixed across all experiments and are not tuned on validation data. For each method, we select the checkpoint that achieves the highest validation accuracy and report performance on the held-out test split using this checkpoint. Test data are never used for hyperparameter selection or model selection.

Evaluation and metrics. Across all datasets, we report classification accuracy as the primary performance metric. For binary classification tasks (SST-2, QQP, and Rotten Tomatoes), we additionally report micro-, macro-, and weighted F_1 scores to account for potential class imbalance. To evaluate ranking quality and probabilistic reliability, we also report ROC-AUC, precision–recall AUC, Matthews correlation coefficient (MCC), balanced accuracy, LogLoss, Brier score, and expected calibration error (ECE). All metrics are computed on the test split using the checkpoint selected based on validation performance.

Experimental setup. Inputs are tokenized using the `bert-base-uncased` WordPiece tokenizer and truncated or padded to a maximum sequence length of 128 tokens. Models use a two-layer Transformer encoder with hidden size 128, two attention heads, feedforward dimension 256, and $m = 256$ random features per head. In our particle-based view, each column of $\mathbf{\Omega}_N \in \mathbb{R}^{H \times d_k \times m}$ is a particle, so this corresponds to 256 particles per head (512 per layer; 1024 particles total across the two-layer encoder). Dropout is set to 0.1, and mean pooling is applied over token representations. We use a $\log$ repulsion force between particles.

Training uses a batch size of 64. When kernel optimization is enabled, training proceeds in two phases. In Phase A, the attention feature map parameters $\mathbf{\Omega}_N$ (and optionally layer normalization and value projection parameters) are optimized using an alignment loss objective for up to 10 epochs via particle optimization (SGLD/Langevin dynamics with step size $\eta = 2 \times 10^{-3}$ and inverse temperature $\beta = 50$, i.e., Gaussian noise scale $\sqrt{2\eta/\beta}$, plus a repulsion term with $\lambda = 10^{-3}$ and gradient clipping at norm 10), while the remaining model parameters are frozen. Column-wise L_2 norms of $\mathbf{\Omega}_N$ (i.e., per-particle norms) are constrained to 1.5. Training in Phase A stops early if the maximum change in $\mathbf{\Omega}_N$ falls below 10^{-6}. In Phase B, the kernel parameters are frozen and the model is trained end-to-end with cross-entropy loss for 10 epochs. For vanilla baselines, only Phase B is used. Optimization in Phase B is performed with Adam using a learning rate of 2×10^{-4}.

Effect of Kernel Learning. Tables 1, 2, and 3 quantify the effect of target-alignment kernel learning across SST-2, QQP, and Rotten Tomatoes. On SST-2 (Table 1), where overall performance is near saturation, kernelization yields feature-map-dependent gains. Kernel-porf-softplus attains the best accuracy and MCC (0.8188/0.6376). Several kernel variants improve both discriminative and probabilistic metrics: for example, Kernel-elu increases accuracy from 0.8028 to 0.8108 while simultaneously reducing LogLoss (0.4938→0.4502) and Brier score (0.1505→0.1392), and Kernel-softmaxfeat improves accuracy (0.8005→0.8085) while achieving near-optimal LogLoss (0.4381). Calibration-specific improvements are also evident: Kernel-cos2 yields the lowest Brier score (0.1366), while Kernel-favor achieves the lowest ECE (0.3086). However, not all kernels are uniformly beneficial—Kernel-softplus degrades both Brier and ECE—highlighting sensitivity to the choice of feature map.

On the more challenging QQP task (Table 2), kernel learning provides more consistent benefits across feature maps. Kernel-softmaxfeat is the strongest $Q{=}K$ model (Acc 0.7909, MCC 0.5506), with substantial reductions in LogLoss (0.4625→0.4354) and Brier score (0.1512→0.1418), while Kernel-favor achieves the best calibration as measured by ECE (0.4106). Although kernelization roughly doubles training time, it does not affect inference-time complexity, indicating that alignment primarily reshapes representation geometry and probabilistic reliability rather than merely sharpening decision boundaries.

Results on Rotten Tomatoes (Table 3) exhibit similar but more modest trends. Kernel-softmaxfeat achieves the highest accuracy and MCC (0.7167/0.4345) while also attaining the lowest LogLoss

(0.5697) and Brier score (0.1929), suggesting that kernel alignment is particularly effective in improving probabilistic calibration on smaller, noisier datasets. As on SST-2, gains vary across feature maps, reinforcing the importance of the kernel choice.

Despite these improvements, Vaswani-style softmax attention remains the strongest performer on QQP. Its advantage stems from explicitly modeling dense all-pairs token interactions through an $\ell \times \ell$ attention matrix, incurring $\mathcal{O}(\ell^2)$ time and memory complexity in the sequence length ℓ. This expressivity is well suited to paraphrase detection, where fine-grained cross-token alignment is critical, but the quadratic scaling limits practicality for longer sequences. In contrast, linear-attention variants trade some modeling capacity for $\mathcal{O}(\ell)$-type scaling (up to the feature dimension), making them preferable when sequence length or throughput constraints dominate.

Appendix A. Appendix

We organize the appendix as follows:

- In Section A.3, we present the proof of Theorem 4.
- In Section A.4 we present the large deviation result of Theorem 6.

A.1. Notation

We define the notation as follows:

- $\mathcal{M}(\Omega)$: The space of measures on the measurable space $(\Omega, \mathcal{B})$, where $\mathcal{B}$ is the Borel σ-algebra on Ω.
- $\hat{\mu} \in \mathcal{M}(\Omega)$: A random (counting) measure, typically representing the realization of the point process. It is a random element in the space of measures on Ω.
- $\mu \in \mathcal{M}(\Omega)$: The deterministic (limiting) measure, often representing the stationary or equilibrium distribution of the point process as $N \to \infty$.
- $\mathcal{P} : \Xi \to \mathcal{M}(\Omega)$: A point process mapping from the probability space $(\Xi, \mathcal{F}, \mathbb{P})$ to the space of measures on Ω.
- $\nu = \mathbb{P} \circ \mathcal{P}^{-1}$: The push-forward measure of the point process, governing the randomness of the point process and describing the distribution of realizations of the point process.
- Λ: The intensity measure, defined as

$$\Lambda(B) = \mathbb{E}_{\mu \sim \nu}[\mu(B)] \quad \text{for} \quad B \in \mathcal{B},$$

  where μ is the random measure associated with a realization of the point process.
- Ξ: The underlying probability space, often taken as a sample space of configurations for the point process.
- $\hat{\mu}(B)$: The counting measure or number of points in a subset $B \in \mathcal{B}$, corresponding to the realization of the random measure $\hat{\mu}$.

- $\mathbb{E}[\cdot]$: Expectation with respect to the probability measure governing the point process. In this context, it is typically the expectation under ν, the push-forward measure.
- $\mu(B)$: The number of points in the set B as determined by the limiting measure μ.
- X: A random variable (or random element) associated with the point process, representing a random realization or observation.
- x: A realization of the random variable X, i.e., an outcome or observation from the random process.

A.2. Proof of Lemma 3

Insert $\phi_{\boldsymbol{\omega}_k,b_k}(\boldsymbol{x}) = \sqrt{2}\cos(\boldsymbol{\omega}_k^\top \boldsymbol{x} + b_k)$ into (B.4):

$$\frac{1}{N}\sum_{k=1}^{N}\phi_{\boldsymbol{\omega}_k,b_k}(\boldsymbol{x}_i)\phi_{\boldsymbol{\omega}_k,b_k}(\boldsymbol{x}_j) = \frac{2}{N}\sum_{k=1}^{N}\cos(\alpha_{ik})\cos(\alpha_{jk}), \quad \alpha_{ik} \stackrel{\text{def}}{=} \boldsymbol{\omega}_k^\top \boldsymbol{x}_i + b_k.$$

Substituting and swapping finite sums gives

$$\mathcal{E}_N(\boldsymbol{\Omega}_N, \boldsymbol{b}) = -\frac{2}{n(n-1)N}\sum_{k=1}^{N}\sum_{i\neq j} y_i y_j \cos(\alpha_{ik})\cos(\alpha_{jk}). \tag{A.1}$$

For any $\boldsymbol{a}, \boldsymbol{y} \in \mathbb{R}^n$,

$$\sum_{i\neq j} y_i y_j a_i a_j = \Big(\sum_i y_i a_i\Big)^2 - \sum_i y_i^2 a_i^2. \tag{A.2}$$

Applying (A.2) with $a_i = \cos(\alpha_{ik})$ yields

$$\sum_{i\neq j} y_i y_j \cos(\alpha_{ik})\cos(\alpha_{jk}) = \big(\boldsymbol{y}^\top \boldsymbol{c}_k\big)^2 - \sum_i y_i^2 \cos^2(\alpha_{ik}), \quad (\boldsymbol{c}_k)_i \stackrel{\text{def}}{=} \cos(\alpha_{ik}).$$

Take expectation over the phases $\boldsymbol{b}$, assumed i.i.d. $\mathrm{Unif}[0, 2\pi]$ and independent of $(\boldsymbol{\Omega}_N, \{\boldsymbol{x}_i\})$:

$$\mathbb{E}_{b_k}\big[\cos^2(\boldsymbol{\omega}_k^\top \boldsymbol{x}_i + b_k)\big] \;=\; \tfrac{1}{2}.$$

Therefore,

$$\mathbb{E}_{b_k}\left[\sum_i y_i^2 \cos^2(\alpha_{ik})\right] \;=\; \tfrac{1}{2}\sum_i y_i^2.$$

In binary classification ($y_i^2 = 1$), this equals $n/2$ and is *independent of* $\boldsymbol{\omega}_k$.

Next, write $u_{ik} \stackrel{\text{def}}{=} \boldsymbol{\omega}_k^\top \boldsymbol{x}_i$ and expand $\cos(u_{ik} + b_k) = \cos u_{ik}\cos b_k - \sin u_{ik}\sin b_k$. Define the vectors $\boldsymbol{u}_k = (\cos u_{1k}, \ldots, \cos u_{nk})^\top$, and $\boldsymbol{v}_k = (\sin u_{1k}, \ldots, \sin u_{nk})^\top$.. Then

$$\boldsymbol{y}^\top \boldsymbol{c}_k = \boldsymbol{y}^\top\big(\cos b_k\, \boldsymbol{u}_k - \sin b_k\, \boldsymbol{v}_k\big) \tag{A.3}$$

and

$$(\boldsymbol{y}^\top \boldsymbol{c}_k)^2 = (\cos b_k)^2(\boldsymbol{y}^\top \boldsymbol{u}_k)^2 + (\sin b_k)^2(\boldsymbol{y}^\top \boldsymbol{v}_k)^2 - 2\cos b_k \sin b_k\,(\boldsymbol{y}^\top \boldsymbol{u}_k)(\boldsymbol{y}^\top \boldsymbol{v}_k). \tag{A.4}$$

Taking expectation over b_k and using $\mathbb{E}[\cos^2 b_k] = \mathbb{E}[\sin^2 b_k] = \frac{1}{2}$ and $\mathbb{E}[\sin b_k \cos b_k] = 0$, we obtain

$$\mathbb{E}_{b_k}\Big[(\boldsymbol{y}^\top \boldsymbol{c}_k)^2\Big] = \tfrac{1}{2}\Big((\boldsymbol{y}^\top \boldsymbol{u}_k)^2 + (\boldsymbol{y}^\top \boldsymbol{v}_k)^2\Big).$$

Combining the two expectations,

$$\mathbb{E}_{b_k}\left[\sum_{i\neq j} y_i y_j \cos(\alpha_{ik})\cos(\alpha_{jk})\right] = \tfrac{1}{2}\Big((\boldsymbol{y}^\top \boldsymbol{u}_k)^2 + (\boldsymbol{y}^\top \boldsymbol{v}_k)^2\Big) - \tfrac{1}{2}\sum_i y_i^2,$$

where the second term is a constant (equal to $n/2$ in the binary case) and thus does not depend on $\boldsymbol{\omega}_k$. Summing over k and inserting into (A.1),

$$\mathbb{E}_{\boldsymbol{b}}\big[\mathcal{E}_N(\boldsymbol{\Omega}_N, \boldsymbol{b})\big] = -\frac{1}{n(n-1)N}\sum_{k=1}^{N}\Big((\boldsymbol{y}^\top \boldsymbol{u}_k)^2 + (\boldsymbol{y}^\top \boldsymbol{v}_k)^2\Big),$$

where the equality is up to an additive constant independent of $\boldsymbol{\Omega}_N$. Finally, stack columns $\boldsymbol{C} = \cos(\boldsymbol{X}\boldsymbol{\Omega}_N^\top) = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_N]$, $\boldsymbol{S} = \sin(\boldsymbol{X}\boldsymbol{\Omega}_N^\top) = [\boldsymbol{v}_1 \cdots \boldsymbol{v}_N]$ to write

$$\sum_{k=1}^{N}\Big((\boldsymbol{y}^\top \boldsymbol{u}_k)^2 + (\boldsymbol{y}^\top \boldsymbol{v}_k)^2\Big) = \|\boldsymbol{y}^\top \boldsymbol{C}\|_2^2 + \|\boldsymbol{y}^\top \boldsymbol{S}\|_2^2.$$

This yield the stated result

$$\mathbb{E}_{\boldsymbol{b}}\big[\mathcal{E}_N(\boldsymbol{\Omega}_N, \boldsymbol{b})\big] \;\equiv\; -\frac{1}{n^2}\left(\big\|\boldsymbol{y}^\top \cos(\boldsymbol{X}\boldsymbol{\Omega}_N^\top)\big\|_2^2 + \big\|\boldsymbol{y}^\top \sin(\boldsymbol{X}\boldsymbol{\Omega}_N^\top)\big\|_2^2\right).$$

A.3. Proof of Theorems 1 and 4

Proof We prove Theorem 4; Theorem 1 follows as the special case $V = V_{\mathcal{D}}$. We present a proof sketch based on the projected-particle mean-field approach. Specifically, we construct a continuous-time interpolation of the projected Langevin particles and compare it with a reflected Itô diffusion. A mean-field (propagation-of-chaos) estimate is then used to replace the empirical drift with the law-dependent drift. Finally, we identify the corresponding adjoint Fokker–Planck equation together with its Robin boundary condition. We describe the comparison step in some detail, as the reflection term is precisely what gives rise to the boundary condition.

Step 1: identification of the mean-field drift. For $\mu \in \mathcal{P}(\Omega)$, define

$$U[\mu](\boldsymbol{\omega}) \overset{\text{def}}{=} V(\boldsymbol{\omega}) + \lambda \int_\Omega g_s(\boldsymbol{\omega} - \boldsymbol{\omega}')\,\mu(d\boldsymbol{\omega}'), \qquad b[\mu](\boldsymbol{\omega}) \overset{\text{def}}{=} -\nabla U[\mu](\boldsymbol{\omega}). \tag{A.5}$$

By compactness of Ω and the assumed smooth regularization of g_s (or the collision-free restriction), there is a constant $L < \infty$, independent of N, such that for all $\boldsymbol{\omega}, \widetilde{\boldsymbol{\omega}} \in \Omega$ and $\mu, \nu \in \mathcal{P}(\Omega)$,

$$\|b[\mu](\boldsymbol{\omega}) - b[\nu](\widetilde{\boldsymbol{\omega}})\| \le L\Big(\|\boldsymbol{\omega} - \widetilde{\boldsymbol{\omega}}\| + W_1(\mu,\nu)\Big) \le L\Big(\|\boldsymbol{\omega} - \widetilde{\boldsymbol{\omega}}\| + W_2(\mu,\nu)\Big). \tag{A.6}$$

Let $\mu_N^{(-k)} = (N-1)^{-1}\sum_{\ell\neq k}\delta_{\boldsymbol{\omega}_\ell}$. Since g_s is even and the finite interaction is normalized as $1/(2N(N-1))\sum_{k\neq\ell} g_s(\boldsymbol{\omega}_k-\boldsymbol{\omega}_\ell)$,

$$-N\nabla_{\boldsymbol{\omega}_k}\mathcal{H}_N(\boldsymbol{\Omega}_N) = -\nabla V(\boldsymbol{\omega}_k) - \frac{\lambda}{N-1}\sum_{\ell\neq k}\nabla g_s(\boldsymbol{\omega}_k-\boldsymbol{\omega}_\ell) = b[\mu_N^{(-k)}](\boldsymbol{\omega}_k). \tag{A.7}$$

Moreover $\|b[\mu_N^{(-k)}](\boldsymbol{\omega}_k) - b[\mu_N](\boldsymbol{\omega}_k)\| \le C/N$. Thus the algorithm is the projected Euler scheme for the empirical McKean drift, up to a uniformly vanishing $O(N^{-1})$ self-interaction error.

Step 2: continuous-time embedding and the reflected diffusion comparison. Construct Brownian motions $\boldsymbol{B}_k$ so that $\boldsymbol{B}_k((m+1)\eta_N) - \boldsymbol{B}_k(m\eta_N) = \sqrt{\eta_N}\,\boldsymbol{\xi}_k^m$. Let $\overline{\boldsymbol{\omega}}_k^{N,\eta}(t) = \boldsymbol{\omega}_k^m$ for $t\in[m\eta_N,(m+1)\eta_N)$. The recursion in (E.2) can be written as

$$\overline{\boldsymbol{\omega}}_k^{N,\eta}(m\eta_N) = \mathcal{P}_{\overline{\Omega}}\Big(\overline{\boldsymbol{\omega}}_k^{N,\eta}((m-1)\eta_N) + \eta_N b[\overline{\mu}^{N,(-k)}_{(m-1)\eta_N}](\overline{\boldsymbol{\omega}}_k^{N,\eta}((m-1)\eta_N)) + \sqrt{2/\beta}\,\Delta\boldsymbol{B}_k^m\Big), \tag{A.8}$$

where $\overline{\mu}_t^{N,(-k)} = (N-1)^{-1}\sum_{\ell\neq k}\delta_{\overline{\boldsymbol{\omega}}_\ell^{N,\eta}(t)}$ and $\Delta\boldsymbol{B}_k^m = \boldsymbol{B}_k(m\eta_N) - \boldsymbol{B}_k((m-1)\eta_N)$. This is the natural cadlag embedding of the projected particle chain.

The limiting continuous reflected particle system associated with (A.8) is

$$d\boldsymbol{X}_k^N(t) = b[\mu_t^{N,(-k)}](\boldsymbol{X}_k^N(t))\,dt + \sqrt{2/\beta}\,d\boldsymbol{B}_k(t) - \boldsymbol{n}(\boldsymbol{X}_k^N(t))\,dL_k^N(t), \qquad \mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{\boldsymbol{X}_j^N(t)}, \tag{A.9}$$

$$\boldsymbol{X}_k^N(t)\in\overline{\Omega}, \qquad L_k^N(t) \text{ is nondecreasing}, \qquad \int_0^T \mathbf{1}_\Omega(\boldsymbol{X}_k^N(t))\,dL_k^N(t) = 0. \tag{A.10}$$

Here $\boldsymbol{n}$ is the outward normal, so $-\boldsymbol{n}\,dL_k^N$ is the inward reflection. The Skorokhod problem on a compact convex C^2 domain has a unique reflected solution, and the projection map in (A.8) is precisely the Euler approximation of this reflected equation. The projected-Euler consistency assumption in the theorem means that, for each fixed T,

$$\varepsilon_{N,\eta}(T) \stackrel{\text{def}}{=} \mathbb{E}\sup_{0\le t\le T}\frac{1}{N}\sum_{k=1}^N \|\overline{\boldsymbol{\omega}}_k^{N,\eta}(t) - \boldsymbol{X}_k^N(t)\|^2 \longrightarrow 0. \tag{A.11}$$

For smooth bounded drifts this follows from the standard Euler–Skorokhod estimate; the condition on η_N makes the discrete projection error negligible on the mean-field scale.

Step 3: nonlinear reflected process and propagation of chaos. Let $\boldsymbol{X}_k(t)$, $k\ge 1$, be i.i.d. copies of the nonlinear reflected McKean–Vlasov process

$$d\boldsymbol{X}_k(t) = b[\mu_t](\boldsymbol{X}_k(t))\,dt + \sqrt{2/\beta}\,d\boldsymbol{B}_k(t) - \boldsymbol{n}(\boldsymbol{X}_k(t))\,dL_k(t), \qquad \mu_t = \mathrm{Law}(\boldsymbol{X}_k(t)), \tag{A.12}$$

$$\boldsymbol{X}_k(t)\in\overline{\Omega}, \qquad \int_0^T \mathbf{1}_\Omega(\boldsymbol{X}_k(t))\,dL_k(t) = 0. \tag{A.13}$$

The Lipschitz bound (A.6) gives existence and uniqueness by a fixed-point argument on measure-valued curves. Couple $\boldsymbol{X}_k^N$ and $\boldsymbol{X}_k$ with the same Brownian motion and the same initial particle. For convex Ω, the reflection map is monotone:

$$\big(\boldsymbol{X}_k^N(t) - \boldsymbol{X}_k(t)\big) \cdot \Big(- \boldsymbol{n}(\boldsymbol{X}_k^N(t))\, dL_k^N(t) + \boldsymbol{n}(\boldsymbol{X}_k(t))\, dL_k(t)\Big) \le 0. \tag{A.14}$$

Applying Itô's formula to $\|\boldsymbol{X}_k^N(t) - \boldsymbol{X}_k(t)\|^2$, using (A.6), and averaging over k yields

$$e_N(t) \stackrel{\text{def}}{=} \mathbb{E} \sup_{0\le r\le t} \frac{1}{N}\sum_{k=1}^N \|\boldsymbol{X}_k^N(r) - \boldsymbol{X}_k(r)\|^2 \tag{A.15}$$

$$\le C_T \int_0^t e_N(s)\, ds + C_T \int_0^t \mathbb{E} W_2^2\left(\frac{1}{N}\sum_{j=1}^N \delta_{\boldsymbol{X}_j(s)}, \mu_s\right) ds + \frac{C_T}{N^2}. \tag{A.16}$$

The last term is the leave-one-out/self-interaction error. Since Ω is compact and $\boldsymbol{X}_j(s)$ are i.i.d. with law μ_s,

$$a_N(T) \stackrel{\text{def}}{=} \sup_{0\le s\le T} \mathbb{E} W_2^2\left(\frac{1}{N}\sum_{j=1}^N \delta_{\boldsymbol{X}_j(s)}, \mu_s\right) \longrightarrow 0. \tag{A.17}$$

Gronwall's inequality therefore gives

$$\sup_{0\le t\le T} \mathbb{E} W_2^2(\mu_t^N, \mu_t) \le C_T\big(a_N(T) + N^{-1} + \varepsilon_{N,\eta}(T)\big) \longrightarrow 0. \tag{A.18}$$

This is the propagation-of-chaos statement: any fixed finite subcollection of particles becomes asymptotically independent, and each coordinate has law μ_t.

Step 4: the Girsanov/change-of-measure ingredient. The preceding synchronous estimate can equivalently be written as a compact-domain change-of-measure estimate. To make this explicit, define

$$\boldsymbol{u}_k(t) = \sqrt{\beta/2}\Big(b[\mu_t^N](\boldsymbol{X}_k^N(t)) - b[\mu_t](\boldsymbol{X}_k^N(t))\Big). \tag{A.19}$$

Novikov's condition holds because Ω is compact and the drift is bounded. Hence the exponential martingale

$$\mathcal{Z}_T = \exp\left(-\sum_{k=1}^N \int_0^T \boldsymbol{u}_k(t)\cdot d\boldsymbol{B}_k(t) - \frac{1}{2}\sum_{k=1}^N \int_0^T \|\boldsymbol{u}_k(t)\|^2 dt\right) \tag{A.20}$$

defines a probability measure under which $\widetilde{\boldsymbol{B}}_k(t) = \boldsymbol{B}_k(t) + \int_0^t \boldsymbol{u}_k(s)ds$ are Brownian motions. The relative entropy of this tilted law with respect to the original law is

$$D_{\mathrm{KL}}(\widetilde{\mathbb{P}}_{N,T}\|\mathbb{P}_{N,T}) = \frac{1}{2}\sum_{k=1}^N \widetilde{\mathbb{E}} \int_0^T \|\boldsymbol{u}_k(t)\|^2 dt \le C\, N \int_0^T \widetilde{\mathbb{E}} W_2^2(\mu_t^N, \mu_t) dt. \tag{A.21}$$

By Pinsker's inequality and the bounded diameter of Ω, this entropy bound gives, at the empirical-measure level,

$$\mathbb{E}W_2^2(\mu_t^{N,\mathrm{emp}}, \mu_t^{N,\mathrm{nl}}) \le \mathrm{diam}(\Omega)^2 \left(\frac{2}{N} D_{\mathrm{KL}}(\widetilde{\mathbb{P}}_{N,t} \| \mathbb{P}_{N,t}) \right)^{1/2}, \tag{A.22}$$

where $\mu_t^{N,\mathrm{emp}}$ and $\mu_t^{N,\mathrm{nl}}$ denote the empirical measures of the empirical-drift and decoupled reflected systems. This quantifies the cost of replacing the particle drift by the law-dependent McKean drift. Combining (A.11), (A.18), (A.22), and the triangle inequality gives

$$\sup_{0 \le t \le T} \mathbb{E}W_2^2 \left(\frac{1}{N} \sum_{k=1}^{N} \delta_{\boldsymbol{\omega}_k^{\lfloor t/\eta_N \rfloor}}, \mu_t \right) \longrightarrow 0. \tag{A.23}$$

Since W_2 convergence on compact Ω implies weak convergence, this proves the empirical-measure convergence in both theorem statements.

Step 5: identification of the McKean–Vlasov PDE and the boundary condition. Let $\psi \in C^2(\overline{\Omega})$ belong to the generator domain of the reflected diffusion, i.e. $\partial_{\boldsymbol{n}}\psi = 0$ on $\partial\Omega$. Applying Itô's formula to $\psi(\boldsymbol{X}_t)$ in (A.12) gives

$$\frac{d}{dt} \int_\Omega \psi(\boldsymbol{\omega})\, \mu_t(d\boldsymbol{\omega}) = \int_\Omega \left[b[\mu_t](\boldsymbol{\omega}) \cdot \nabla\psi(\boldsymbol{\omega}) + \beta^{-1}\Delta\psi(\boldsymbol{\omega}) \right] \mu_t(d\boldsymbol{\omega}). \tag{A.24}$$

The local-time term is $-\partial_{\boldsymbol{n}}\psi(\boldsymbol{X}_t)dL_t$, and therefore vanishes for this generator domain. If $\mu_t(d\boldsymbol{\omega}) = \rho_t(\boldsymbol{\omega})d\boldsymbol{\omega}$, $b[\mu_t] = -\nabla U_t$, and $U_t = U[\mu_t]$, (A.24) becomes

$$\frac{d}{dt} \int_\Omega \psi\rho_t\, d\boldsymbol{\omega} = \int_\Omega \left[-\nabla U_t \cdot \nabla\psi + \beta^{-1}\Delta\psi \right] \rho_t\, d\boldsymbol{\omega}. \tag{A.25}$$

The adjoint of the reflected generator is therefore

$$\partial_t \rho_t = -\nabla \cdot (\rho_t b[\mu_t]) + \beta^{-1}\Delta\rho_t = \nabla \cdot (\rho_t \nabla U_t) + \beta^{-1}\Delta\rho_t \qquad \text{in } \Omega. \tag{A.26}$$

To identify the boundary condition, integrate the last display against arbitrary smooth test functions and use Green's formula. The boundary contribution is

$$\int_{\partial\Omega} \psi(\boldsymbol{\omega}) \Big(\rho_t(\boldsymbol{\omega}) b[\mu_t](\boldsymbol{\omega}) - \beta^{-1}\nabla\rho_t(\boldsymbol{\omega}) \Big) \cdot \boldsymbol{n}(\boldsymbol{\omega})\, dS(\boldsymbol{\omega}). \tag{A.27}$$

Reflection means that the probability current through $\partial\Omega$ vanishes. Since $b[\mu_t] = -\nabla U_t$, this current condition is

$$\Big(\rho_t \nabla U_t + \beta^{-1}\nabla\rho_t \Big) \cdot \boldsymbol{n} = 0 \qquad \text{on } (0,T] \times \partial\Omega. \tag{A.28}$$

For smooth ρ_t, (A.28) is exactly the Robin form

$$\partial_{\boldsymbol{n}}\rho_t + \beta\rho_t\partial_{\boldsymbol{n}}U_t = 0 \qquad \text{on } (0,T] \times \partial\Omega, \tag{A.29}$$

which is exactly the Robin/no-flux boundary condition generated by the reflected dynamics. The initial condition follows from $\mu_0^N \Rightarrow \rho_0 d\boldsymbol{\omega}$. Taking $\psi \equiv 1$ in the weak formulation gives $\int_\Omega \rho_t = 1$, and nonnegativity follows because ρ_t is the density of the law of the reflected process. Finally, sending $\beta \to \infty$ removes the diffusion term and reduces the no-flux condition to $(\rho_t \boldsymbol{v}_t) \cdot \boldsymbol{n} = 0$, where $\boldsymbol{v}_t = -\nabla U_t$, yielding the deterministic reflected continuity equation. ■

A.4. Proof of Theorem 6

The proof of Theorem 6 follows standard application of Varadhan's lemma [14]. We provide the proof in multiple steps.

A.4.1. REWRITE THE HAMILTONIAN AS A FUNCTIONAL OF μ_N

Let ν denote Lebesgue measure on Ω and set

$$\rho \stackrel{\text{def}}{=} \frac{\nu}{\nu(\Omega)}.$$

We use ρ as reference probability measure, i.e., $\rho \in \mathcal{P}(\Omega)$. We also recall the definition of empirical measure $\mu_N = \frac{1}{N}\sum_{k=1}^{N} \delta_{\boldsymbol{\omega}_k}$. Under the product measure $\rho^{\otimes N}$ on Ω^N, the empirical measure μ_N satisfies Sanov's theorem: it obeys an LDP on $\mathcal{P}(\Omega)$ with speed N and good rate function

$$I(\mu) = \mathrm{Ent}(\mu \mid \rho). \tag{A.30}$$

Now, define the bounded measurable function

$$V(\boldsymbol{\omega}) \stackrel{\text{def}}{=} -\frac{1}{n(n-1)} \sum_{0 \le i \ne j \le n} y_i y_j \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j). \tag{A.31}$$

Then by the Monte–Carlo substitution in (B.5) and the definition (B.4), the empirical loss is exactly a linear functional of μ_N:

$$\mathcal{E}_N(\boldsymbol{\Omega}_N) = -\frac{1}{n(n-1)} \sum_{i \ne j} y_i y_j \int_\Omega \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\, \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j)\, \mu_N(\mathrm{d}\boldsymbol{\omega}) = \int_\Omega V(\boldsymbol{\omega})\, \mu_N(\mathrm{d}\boldsymbol{\omega}), \tag{A.32}$$

Assume the interaction term is of mean-field form (as in the Gibbs law (E.7)):

$$\mathcal{W}_{N,s}(\boldsymbol{\Omega}_N) \stackrel{\text{def}}{=} \frac{1}{2N(N-1)} \sum_{1 \le k \ne \ell \le N} g_s(\boldsymbol{\omega}_k - \boldsymbol{\omega}_\ell). \tag{A.33}$$

Define the corresponding continuum interaction functional on $\mathcal{P}(\Omega)$:

$$\mathcal{W}_s(\mu) \stackrel{\text{def}}{=} \frac{1}{2} \int_\Omega \int_\Omega g_s(\boldsymbol{\omega} - \boldsymbol{\omega}')\, \mu(\mathrm{d}\boldsymbol{\omega})\mu(\mathrm{d}\boldsymbol{\omega}'). \tag{A.34}$$

Then the full empirical Hamiltonian in (B.3) can be written as

$$\mathcal{H}_{N,s}(\boldsymbol{\Omega}_N) = \mathcal{E}_N(\boldsymbol{\Omega}_N) + \lambda\, \mathcal{W}_{N,s}(\boldsymbol{\Omega}_N). \tag{A.35}$$

With this normalization, $\mathcal{W}_{N,s}(\boldsymbol{\Omega}_N)$ converges to $\mathcal{W}_s(\mu_N)$; after truncation the difference is only a harmless finite-size correction. To handle possible singularities of the kernel g_s at $\mathbf{0}$, we introduce a truncation.

A.4.2. TRUNCATION OF THE INTERACTION KERNEL AND REMOVAL OF THE DIAGONAL CONSTANT

The Columb/Riesz kernel g_s is singular at the origin. In order to work with bounded continuous functionals on $\mathcal{P}(\Omega)$ (so that Varadhan's lemma applies), we introduce a truncation. For $\varepsilon \geq 0$ define the truncated kernel

$$g_s^\varepsilon(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \min\{\|\boldsymbol{\omega}\|^{-s}, \varepsilon^{-s}\} \tag{A.36}$$

and adopt the convention that $g_s^\varepsilon(\mathbf{0})$ is the (finite) value of this truncation at $\mathbf{0}$. In particular, $g_s^\varepsilon(\mathbf{0}) = \varepsilon^{-s}$. Define the truncated empirical interaction by

$$\mathcal{W}_{N,s}^\varepsilon(\Omega_N) \stackrel{\text{def}}{=} \frac{1}{2N(N-1)} \sum_{1\leq k\neq \ell\leq N} g_s^\varepsilon(\boldsymbol{\omega}_k - \boldsymbol{\omega}_\ell), \tag{A.37}$$

and the corresponding truncated mean-field interaction functional on $\mathcal{P}(\Omega)$ by

$$\mathcal{W}_s^\varepsilon(\mu) \stackrel{\text{def}}{=} \frac{1}{2} \int_\Omega \int_\Omega g_s^\varepsilon(\boldsymbol{\omega} - \boldsymbol{\omega}')\, \mu(d\boldsymbol{\omega})\, \mu(d\boldsymbol{\omega}'). \tag{A.38}$$

For fixed ε, the map $\mu \mapsto \mathcal{W}_s^\varepsilon(\mu)$ is continuous on $\mathcal{P}(\Omega)$ since g_s^ε is bounded and continuous on Ω_N. Define the truncated energy functional on $\mathcal{P}(\Omega)$ by

$$\mathcal{H}_s^\varepsilon(\mu) \stackrel{\text{def}}{=} \int_\Omega V(\boldsymbol{\omega})\, \mu(d\boldsymbol{\omega}) + \lambda\, \mathcal{W}_s^\varepsilon(\mu). \tag{A.39}$$

and define the truncated empirical Hamiltonian by

$$\mathcal{H}_{N,s}^\varepsilon(\boldsymbol{\Omega}_N) \stackrel{\text{def}}{=} \mathcal{E}_N(\boldsymbol{\Omega}_N) + \lambda\, \mathcal{W}_{N,s}^\varepsilon(\boldsymbol{\Omega}_N). \tag{A.40}$$

Lemma 7 (Finite-size correction for empirical measures) *For every configuration $\boldsymbol{\Omega}_N = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N) \in \Omega^N$,*

$$\mathcal{W}_s^\varepsilon(\mu_N) = \frac{N-1}{N} \mathcal{W}_{N,s}^\varepsilon(\boldsymbol{\Omega}_N) + \frac{g_s^\varepsilon(\mathbf{0})}{2N}. \tag{A.41}$$

Equivalently,

$$\mathcal{H}_{N,s}^\varepsilon(\boldsymbol{\Omega}_N) = \mathcal{H}_s^\varepsilon(\mu_N) + \frac{\lambda}{N-1} \mathcal{W}_s^\varepsilon(\mu_N) - \lambda\, \frac{g_s^\varepsilon(\mathbf{0})}{2(N-1)}. \tag{A.42}$$

In particular, for fixed $\varepsilon > 0$,

$$\sup_{\mu\in\mathcal{P}(\Omega)} \left| \frac{\lambda}{N-1} \mathcal{W}_s^\varepsilon(\mu) \right| \leq \frac{|\lambda|\, \varepsilon^{-s}}{2(N-1)}.$$

Thus replacing $\mathcal{H}_{N,s}^\varepsilon(\boldsymbol{\Omega}_N)$ by $\mathcal{H}_s^\varepsilon(\mu_N)$ changes the normalized logarithmic Laplace limits below by $o(1)$ at speed N when $\beta_N/N \to 1$ and by $o(1)$ at speed β_N when $\beta_N/N \to \infty$.

Proof Since $\mu_N = \frac{1}{N}\sum_{k=1}^N \delta_{\boldsymbol{\omega}_k}$, we have

$$\mathcal{W}_s^\varepsilon(\mu_N) = \frac{1}{2}\frac{1}{N^2} \sum_{k,\ell=1}^N g_s^\varepsilon(\boldsymbol{\omega}_k - \boldsymbol{\omega}_\ell) = \frac{1}{2}\frac{1}{N^2} \sum_{k\neq\ell} g_s^\varepsilon(\boldsymbol{\omega}_k - \boldsymbol{\omega}_\ell) + \frac{g_s^\varepsilon(\mathbf{0})}{2N}.$$

The off-diagonal term equals

$$\frac{N-1}{N}\,\frac{1}{2N(N-1)}\sum_{k\neq\ell} g_s^{\varepsilon}(\boldsymbol{\omega}_k-\boldsymbol{\omega}_\ell)=\frac{N-1}{N}\mathcal{W}_{N,s}^{\varepsilon}(\boldsymbol{\Omega}_N),$$

which proves (A.41). Solving this identity for $\mathcal{W}_{N,s}^{\varepsilon}$ and adding the linear term $\mathcal{E}_N=\int V\,d\mu_N$ gives (A.42). The uniform bound follows from $0\le g_s^{\varepsilon}\le\varepsilon^{-s}$. The deterministic constant in (A.42) is absorbed into the partition function. The remaining tilt is uniformly $O(1/N)$; after multiplication by β_N and division by the relevant speed, its contribution is $O(1/N)$ both in the thermal scale $\beta_N\sim N$ and in the zero-temperature scale β_N. ■

Lemma 7 shows that the $N(N-1)$-normalized Hamiltonian and the mean-field functional $\mathcal{H}_s^{\varepsilon}(\mu_N)$ have the same logarithmic Laplace limits at the speeds used below. The diagonal constant is absorbed by the partition function, and the remaining finite-size tilt is uniformly negligible. In what follows we therefore use the asymptotically equivalent representation

$$\mathbb{P}_{N,\beta_N}^{\varepsilon}(\mathrm{d}\boldsymbol{\Omega}_N)\asymp\frac{1}{\widetilde{Z}_{N,\beta_N}^{\varepsilon}}\exp\big(-\beta_N\,\mathcal{H}_s^{\varepsilon}(\mu_N)\big)\,\rho^{\otimes N}(\mathrm{d}\boldsymbol{\Omega}_N),$$

where $\asymp$ denotes equivalence of the normalized logarithmic Laplace limits at the relevant speed.

A.4.3. CASE $\beta_N/N\to 1$

Assume $\beta_N/N\to 1$ and write $\beta_N=N\alpha_N$ with $\alpha_N=O_N(1)$. We can rewrite the truncated Gibbs law as a tilt by $\mathcal{H}_s^{\varepsilon}(\mu_N)$:

$$\mathbb{P}_{N,\beta_N}^{\varepsilon}(d\Omega_N)\propto\exp\big(-N\alpha_N\,\mathcal{H}_s^{\varepsilon}(\mu_N)\big)\,\rho^{\otimes N}(\mathrm{d}\boldsymbol{\Omega}_N). \tag{A.43}$$

Let $F:\mathcal{P}(\Omega)\to\mathbb{R}$ be bounded and continuous. We leverage Varadhan's lemma [14, p. 51], a rigorous formulation of the Laplace principle (or the saddle point technique) applied to measures satisfying a large deviations property:

Lemma 8 (Varadhan's Lemma [14]) *Suppose a sequence $\{Q_N\}_{N=1}^{\infty}$ of probability measures on $\mathcal{X}$ satisfies a large deviations property with rate function $I(x)$. Let $F:\mathcal{X}\to\mathbb{R}$ be a continuous function that satisfies the tail condition*

$$\lim_{L\to\infty}\limsup_{N\to\infty}\frac{1}{N}\log\int_{x:F(x)\ge L}\exp\big(NF(x)\big)\,Q_N(dx)=-\infty. \tag{A.44}$$

Then

$$\lim_{N\to\infty}\frac{1}{N}\log\int_{\mathcal{X}}\exp\big(NF(x)\big)\,Q_N(dx)=\sup_{x\in\mathcal{X}}\big\{F(x)-I(x)\big\}. \tag{A.45}$$

To apply Varadhan's lemma, we must verify the tail condition. Let

$$Q_N\overset{\text{def}}{=}\rho^{\otimes N}\circ\mu_N^{-1}$$

denote the push-forward law of the empirical measure μ_N on $\mathcal{P}(\Omega)$ under $\rho^{\otimes N}$. Concretely, for any measurable $A \subset \mathcal{P}(\Omega)$,

$$Q_N(A) = \rho^{\otimes N}\big(\{\mathbf{\Omega}_N : \mu_N(\mathbf{\Omega}_N) \in A\}\big).$$

The tail condition in Varadhan's lemma for a functional $\Phi : \mathcal{P}(\Omega) \to \mathbb{R}$ reads

$$\lim_{L\to\infty} \limsup_{N\to\infty} \frac{1}{N} \log \int_{\{\mu:\, \Phi(\mu)\geq L\}} e^{N\Phi(\mu)}\, Q_N(d\mu) \;=\; -\infty. \tag{A.46}$$

Equivalently, since μ_N has law Q_N under $\rho^{\otimes N}$,

$$\lim_{L\to\infty} \limsup_{N\to\infty} \frac{1}{N} \log \mathbb{E}_{\rho^{\otimes N}}\Big[e^{N\Phi(\mu_N)}\mathbf{1}_{\{\Phi(\mu_N)\geq L\}}\Big] \;=\; -\infty. \tag{A.47}$$

In our application, we use the continuous functional for numerator

$$\Phi_N^{\varepsilon}(\mu) \;\stackrel{\text{def}}{=}\; -\Big(\alpha_N\, \mathcal{H}_s^{\varepsilon}(\mu) + F(\mu)\Big), \qquad \alpha_N = \beta_N/N \to 1, \tag{A.48}$$

and for the denominator the functional

$$\Psi_N^{\varepsilon}(\mu) \;\stackrel{\text{def}}{=}\; -\alpha_N\, \mathcal{H}_s^{\varepsilon}(\mu). \tag{A.49}$$

We verify (A.46) for Φ_N^{ε}. The same argument applies to Ψ_N^{ε}.

To do so, we first show that $\mathcal{H}_s^{\varepsilon}$ is bounded for fixed $\varepsilon > 0$. Recall

$$\mathcal{H}_s^{\varepsilon}(\mu) = \int_{\Omega} V(\boldsymbol{\omega})\, \mu(\mathrm{d}\boldsymbol{\omega}) \;+\; \lambda\, \mathcal{W}_s^{\varepsilon}(\mu), \quad \text{with } \mathcal{W}_s^{\varepsilon}(\mu) = \frac{1}{2}\iint g_s^{\varepsilon}(\boldsymbol{\omega}-\boldsymbol{\omega}')\, \mu(\mathrm{d}\boldsymbol{\omega})\mu(\mathrm{d}\boldsymbol{\omega}').$$

Since V is bounded, for all $\mu \in \mathcal{P}(\Omega)$,

$$\begin{aligned}
\Big|\int_{\Omega} V\, \mathrm{d}\mu\Big| &\leq \|V\|_{\infty} && \text{(A.50)}\\
&= \frac{1}{n(n-1)} \Bigg\| \sum_{0\leq i\neq j\leq n} y_i y_j \phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j) \Bigg\|_{\infty} && \text{(A.51)}\\
&\leq |y_i y_j|\, \|\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_i)\|_{\infty} \|\phi_{\boldsymbol{\omega}}(\boldsymbol{x}_j)\|_{\infty} && \text{(A.52)}\\
&\leq L_{\phi}^2. && \text{(A.53)}
\end{aligned}$$

Moreover, by definition of truncated kernel $g_s^{\varepsilon}(\boldsymbol{\omega})$ in Eq. (A.36), we have $g_s^{\varepsilon} \leq \varepsilon^{-s}$. Therefore, for any $\mu \in \mathcal{P}(\Omega)$,

$$|\mathcal{W}_s^{\varepsilon}(\mu)| = \frac{1}{2}\left|\iint g_s^{\varepsilon}(\omega-\omega')\, \mu(d\omega)\mu(d\omega')\right| \leq \frac{1}{2}\left|\iint \varepsilon^{-s}\, \mu(d\omega)\mu(d\omega')\right| = \frac{\varepsilon^{-s}}{2}. \tag{A.54}$$

Combining (A.51)–(A.54) gives, for all μ,

$$|\mathcal{H}_s^{\varepsilon}(\mu)| \leq L_{\phi}^2 + |\lambda|\, \frac{\varepsilon^{-s}}{2} \;\stackrel{\text{def}}{=}\; C_{\lambda,\varepsilon}. \tag{A.55}$$

Since F is bounded and continuous, define

$$\|F\|_\infty \stackrel{\text{def}}{=} \sup_{\mu\in\mathcal{P}(\Omega)} |F(\mu)| < \infty.$$

Moreover, since $\alpha_N = \mathcal{O}_N(1)$, there exists N_0 and constant $C > 0$ such that for all $N \geq N_0$,

$$|\alpha_N| \leq C. \tag{A.56}$$

Then for all $N \geq N_0$ and all $\mu \in \mathcal{P}(\Omega)$,

$$\begin{aligned} |\Phi_N(\mu)| &= |-\alpha_N \mathcal{H}_s^\varepsilon(\mu) - F(\mu)| && \text{(A.57)}\\ &\leq |\alpha_N|\,|\mathcal{H}_s^\varepsilon(\mu)| + |F(\mu)| && \text{(A.58)}\\ &\leq CC_{\lambda,\varepsilon} + \|F\|_\infty \stackrel{\text{def}}{=} K_{\varepsilon,F}, && \text{(A.59)} \end{aligned}$$

where we used (A.55) and (A.56). Thus, Φ_N is bounded above by $K_{\varepsilon,F}$ uniformly in μ, for all $N \geq N_0$. Similarly, for the denominator functional,

$$\Psi_N(\mu) = -\alpha_N \mathcal{H}_s^\varepsilon(\mu) \leq |\alpha_N|\,|\mathcal{H}_s^\varepsilon(\mu)| \leq CC_\varepsilon \stackrel{\text{def}}{=} K_\varepsilon, \qquad N \geq N_0. \tag{A.60}$$

Fix any $L > K_{M,F}$. Then by (A.59), for all $N \geq N_0$,

$$\{\mu \in \mathcal{P}(\Omega) : \Phi_N(\mu) \geq L\} = \varnothing.$$

Hence the tail integral vanishes, i.e.,

$$\int_{\{\mu:\Phi_N(\mu)\geq L\}} e^{N\Phi_N(\mu)}\, Q_N(d\mu) = 0, \qquad N \geq N_0.$$

Therefore,

$$\frac{1}{N}\log \int_{\{\mu:\Phi_N(\mu)\geq L\}} e^{N\Phi_N(\mu)}\, Q_N(d\mu) = -\infty, \qquad N \geq N_0.$$

This establishes the tail condition (A.46).

$$\lim_{L\to\infty} \limsup_{N\to\infty} \frac{1}{N}\log \int_{\{\mu:\Phi_N(\mu)\geq L\}} e^{N\Phi_N(\mu)}\, Q_N(d\mu) = -\infty, \tag{A.61}$$

The same reasoning, using (A.60), shows that the tail condition also holds for the denominator functional Ψ_N.

Now, by Varadhan's lemma applied to Sanov's LDP (A.30), we obtain

$$\lim_{N\to\infty} \frac{1}{N}\log \mathbb{E}_{\rho^{\otimes N}}\Big[\exp\big(-N(\alpha_N\mathcal{H}_s^\varepsilon + F)(\mu_N)\big)\Big] = -\inf_{\mu\in\mathcal{P}(\Omega)} \Big(\mathcal{H}_s^\varepsilon(\mu) + F(\mu) + \mathrm{Ent}(\mu \mid \rho)\Big), \tag{A.62}$$

$$\lim_{N\to\infty} \frac{1}{N}\log \mathbb{E}_{\rho^{\otimes N}}\Big[\exp\big(-N\alpha_N\mathcal{H}_s^\varepsilon(\mu_N)\big)\Big] = -\inf_{\mu\in\mathcal{P}(\Omega)} \Big(\mathcal{H}_s^\varepsilon(\mu) + \mathrm{Ent}(\mu \mid \rho)\Big). \tag{A.63}$$

Subtracting (A.63) from (A.62) yields the Laplace principle for μ_N under $\mathbb{P}^{\varepsilon}_{N,\beta_N}$ at speed N:

$$\lim_{N\to\infty} -\frac{1}{N}\log \mathbb{E}_{\mathbb{P}^{\varepsilon}_{N,\beta_N}}\big[e^{-NF(\mu_N)}\big] = \inf_{\mu}\Big(F(\mu) + \mathcal{J}^{\varepsilon}_s(\mu)\Big),$$

with

$$\mathcal{J}^{\varepsilon}_s(\mu) = \Big(\mathcal{E}^{\varepsilon}_s(\mu) + \mathrm{Ent}(\mu \mid \rho)\Big) - \inf_{\nu\in\mathcal{P}(\Omega)}\Big(\mathcal{E}^{\varepsilon}_s(\nu) + \mathrm{Ent}(\nu \mid \rho)\Big). \tag{A.64}$$

Since $\mathcal{P}(\Omega)$ is a Polish space and the Laplace principle holds for all $F \in C_b(\mathcal{P}(\Omega))$, Bryc's inverse Varadhan lemma (see, e.g., [11, Theorem 4.4.13]) implies that $(\mu_N)_{N\geq 1}$ satisfies a large deviation principle with speed N and good rate function $\mathcal{J}^{\varepsilon}_s$.

Finally let $\varepsilon \downarrow 0$. Because $g^{\varepsilon}_s \uparrow g_s$, we have $\mathcal{H}^{\varepsilon}_s(\mu) \to \mathcal{H}_s(\mu)$ pointwise (possibly $+\infty$), so $\mathcal{H}_s$ is lower semi-continuous as a supremum of continuous functions. Standard monotone truncation arguments for Laplace principles yield the same Laplace limit with $\mathcal{H}_s$ in place of $\mathcal{H}^{\varepsilon}_s$, and hence the LDP with rate

$$\mathcal{J}_s(\mu) = \Big(\mathcal{H}_s(\mu) + \mathrm{Ent}(\mu \mid \rho)\Big) - \inf_{\gamma}\Big(\mathcal{H}_s(\gamma) + \mathrm{Ent}(\gamma \mid \rho)\Big).$$

To convert to $\mathrm{Ent}(\cdot \mid \nu)$, note that $\mathrm{Ent}(\mu \mid \rho) = \mathrm{Ent}(\mu \mid \nu) + \log \nu(\Omega)$, and the additive constant cancels when subtracting the infimum. This gives the first bullet of the theorem.

A.4.4. CASE $\beta_N/N \to \infty$

Fix $\varepsilon \geq 0$ and recall that under the equivalent representation (Lemma 7),

$$\mathbb{P}^{\varepsilon}_{N,\beta_N}(\mathrm{d}\mathbf{\Omega}_N) = \frac{1}{\widetilde{Z}^{\varepsilon}_{N,\beta_N}} \exp\big(-\beta_N\, \mathcal{H}^{\varepsilon}_s(\mu_N)\big)\, \rho^{\otimes N}(\mathrm{d}\mathbf{\Omega}_N),$$

where the partition function is

$$\widetilde{Z}^{\varepsilon}_{N,\beta_N} \overset{\text{def}}{=} \mathbb{E}_{\rho^{\otimes N}}\Big[\exp\big(-\beta_N\, \mathcal{H}^{\varepsilon}_s(\mu_N)\big)\Big].$$

As before, let

$$Q_N \overset{\text{def}}{=} \rho^{\otimes N} \circ \mu_N^{-1}$$

denote the law of μ_N on $\mathcal{P}(\Omega)$ under the reference measure $\rho^{\otimes N}$.

We notice that Sanov probabilities are negligible at speed $\beta_N/N \to \infty$. The key point is that Q_N satisfies an LDP at speed N, so any Q_N-probability is at worst $\exp(-cN)$, which is negligible on the scale $\beta_N \gg N$.

Lemma 9 (Sanov scale is negligible at speed β_N) *Assume $\beta_N/N \to \infty$. Then for every nonempty open set $U \subset \mathcal{P}(\Omega)$,*

$$\lim_{N\to\infty} \frac{1}{\beta_N}\log Q_N(U) = 0.$$

Proof Since $Q_N(U) \leq 1$, we have $\limsup_{N\to\infty} \frac{1}{\beta_N} \log Q_N(U) \leq 0$.

For the lower bound, note that ρ has full support on Ω and the set $\{\mu \in \mathcal{P}(\Omega) : \mathrm{Ent}(\mu \mid \rho) < \infty\}$ is dense in $\mathcal{P}(\Omega)$ for the weak topology, so any nonempty open U contains some μ with finite entropy. Hence

$$\inf_{\mu\in U} \mathrm{Ent}(\mu \mid \rho) < \infty.$$

By the *lower bound* in Sanov's theorem, for open U,

$$\liminf_{N\to\infty} \frac{1}{N} \log Q_N(U) \;\geq\; -\inf_{\mu\in U} \mathrm{Ent}(\mu \mid \rho) \;>\; -\infty.$$

Therefore,

$$\liminf_{N\to\infty} \frac{1}{\beta_N} \log Q_N(U) = \liminf_{N\to\infty} \frac{N}{\beta_N} \cdot \frac{1}{N} \log Q_N(U) \;\geq\; 0,$$

because $N/\beta_N \to 0$ and $\frac{1}{N} \log Q_N(U)$ is bounded below along a subsequence by a finite constant. Combining with the $\limsup \leq 0$ gives the desired limit 0. ■

Now, define the (truncated) ground state energy

$$e_\varepsilon^\star \;\stackrel{\text{def}}{=}\; \inf_{\mu\in\mathcal{P}(\Omega)} \mathcal{H}_s^\varepsilon(\mu).$$

Since g_s^ε and V are bounded, $\mathcal{H}_s^\varepsilon$ is bounded and continuous on $\mathcal{P}(\Omega)$, hence $e_\varepsilon^\star \in \mathbb{R}$.

Lemma 10 (Partition function at scale β_N) *Under $\beta_N/N \to \infty$,*

$$\lim_{N\to\infty} \frac{1}{\beta_N} \log \widetilde{Z}_{N,\beta_N}^\varepsilon = -\, e_\varepsilon^\star.$$

Proof *Upper bound.* Since $\mathcal{H}_s^\varepsilon(\mu) \geq e_\varepsilon^\star$ for all μ,

$$\widetilde{Z}_{N,\beta_N}^\varepsilon = \mathbb{E}_{\rho^{\otimes N}}\Big[e^{-\beta_N \mathcal{H}_s^\varepsilon(\mu_N)}\Big] \leq e^{-\beta_N e_\varepsilon^\star},$$

so $\limsup_{N\to\infty} \frac{1}{\beta_N} \log \widetilde{Z}_{N,\beta_N}^\varepsilon \leq -e_\varepsilon^\star$.

Lower bound. Fix $\varepsilon > 0$ and choose $\mu_\varepsilon \in \mathcal{P}(\Omega)$ such that $\mathcal{H}_s^\varepsilon(\mu_\varepsilon) \leq e_\varepsilon^\star + \varepsilon$. By continuity of $\mathcal{H}_s^\varepsilon$ there exists a nonempty open neighborhood U of μ_ε such that

$$\sup_{\mu\in U} \mathcal{H}_s^\varepsilon(\mu) \leq e_\varepsilon^\star + 2\varepsilon.$$

Then

$$\widetilde{Z}_{N,\beta_N}^\varepsilon \geq \mathbb{E}_{\rho^{\otimes N}}\Big[\mathbf{1}_{\{\mu_N\in U\}}\, e^{-\beta_N \mathcal{H}_s^\varepsilon(\mu_N)}\Big] \geq e^{-\beta_N (e_\varepsilon^\star + 2\varepsilon)}\, Q_N(U).$$

Taking $\log$ and dividing by β_N yields

$$\frac{1}{\beta_N} \log \widetilde{Z}_{N,\beta_N}^\varepsilon \geq -(e_\varepsilon^\star + 2\varepsilon) + \frac{1}{\beta_N} \log Q_N(U).$$

By Lemma 9, $\frac{1}{\beta_N}\log Q_N(U) \to 0$, hence

$$\liminf_{N\to\infty} \frac{1}{\beta_N}\log \widetilde{Z}^{\varepsilon}_{N,\beta_N} \geq -(e^{\star}_{\varepsilon} + 2\varepsilon).$$

Letting $\varepsilon \downarrow 0$ gives the lower bound $\geq -e^{\star}_{M}$ and completes the proof. ■

Let $F \subset \mathcal{P}(\Omega)$ be closed. Using the ratio representation,

$$\mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N \in F) = \frac{\mathbb{E}_{\rho^{\otimes N}}\big[\mathbf{1}_{\{\mu_N\in F\}}\exp\left(-\beta_N \mathcal{H}^{\varepsilon}_s(\mu_N)\right)\big]}{\widetilde{Z}^{\varepsilon}_{N,\beta_N}}.$$

On the event $\{\mu_N \in F\}$ we have $\exp\big(-\beta_N\mathcal{H}^{\varepsilon}_s(\mu_N)\big) \leq \exp\big(-\beta_N \inf_{\mu\in F}\mathcal{H}^{\varepsilon}_s(\mu)\big)$, hence

$$\mathbb{E}_{\rho^{\otimes N}}\big[\mathbf{1}_{\{\mu_N\in F\}}\exp\left(-\beta_N \mathcal{H}^{\varepsilon}_s(\mu_N)\right)\big] \leq \exp\Big(-\beta_N \inf_{\mu\in F}\mathcal{H}^{\varepsilon}_s(\mu)\Big).$$

Therefore,

$$\limsup_{N\to\infty}\frac{1}{\beta_N}\log \mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N\in F) \leq -\inf_{\mu\in F}\mathcal{H}^{\varepsilon}_s(\mu) - \liminf_{N\to\infty}\frac{1}{\beta_N}\log\widetilde{Z}^{\varepsilon}_{N,\beta_N}.$$

By Lemma 10, $\lim_N \frac{1}{\beta_N}\log\widetilde{Z}^{\varepsilon}_{N,\beta_N} = -e^{\star}_{\varepsilon}$, so

$$\limsup_{N\to\infty}\frac{1}{\beta_N}\log \mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N\in F) \leq -\Big(\inf_{\mu\in F}\mathcal{H}^{\varepsilon}_s(\mu) - e^{\star}_{\varepsilon}\Big). \tag{A.65}$$

Now, let $G \subset \mathcal{P}(\Omega)$ be open and fix $\mu \in G$. By continuity of $\mathcal{H}^{\varepsilon}_s$, there exists a nonempty open neighborhood $U \subset G$ of μ such that

$$\sup_{\nu\in U}\mathcal{H}^{\varepsilon}_s(\nu) \leq \mathcal{H}^{\varepsilon}_s(\mu) + \varepsilon.$$

Then

$$\begin{aligned}
\mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N\in G) &\geq \mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N\in U) && \text{(A.66)}\\
&= \frac{\mathbb{E}_{\rho^{\otimes N}}\big[\mathbf{1}_{\{\mu_N\in U\}}e^{-\beta_N\mathcal{H}^{\varepsilon}_s(\mu_N)}\big]}{\widetilde{Z}^{\varepsilon}_{N,\beta_N}} && \text{(A.67)}\\
&\geq \frac{e^{-\beta_N(\mathcal{H}^{\varepsilon}_s(\mu)+\varepsilon)}\,Q_N(U)}{\widetilde{Z}^{\varepsilon}_{N,\beta_N}}. && \text{(A.68)}
\end{aligned}$$

Using the simple bound $\widetilde{Z}^{\varepsilon}_{N,\beta_N} \leq e^{-\beta_N e^{\star}_{\varepsilon}}$ we obtain

$$\mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N\in G) \geq \exp\Big(-\beta_N(\mathcal{H}^{\varepsilon}_s(\mu) - e^{\star}_{\varepsilon} + \varepsilon)\Big)\,Q_N(U).$$

Taking $\log$ and dividing by β_N gives

$$\liminf_{N\to\infty}\frac{1}{\beta_N}\log\mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N\in G) \geq -(\mathcal{H}^{\varepsilon}_s(\mu) - e^{\star}_{\varepsilon} + \varepsilon) + \liminf_{N\to\infty}\frac{1}{\beta_N}\log Q_N(U).$$

By Lemma 9, the last term is 0, hence

$$\liminf_{N\to\infty} \frac{1}{\beta_N} \log \mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N \in G) \geq -(\mathcal{H}^{\varepsilon}_s(\mu) - e^{\star}_{\varepsilon} + \varepsilon).$$

Letting $\varepsilon \downarrow 0$ and then taking the infimum over $\mu \in G$ yields

$$\liminf_{N\to\infty} \frac{1}{\beta_N} \log \mathbb{P}^{\varepsilon}_{N,\beta_N}(\mu_N \in G) \geq -\Big(\inf_{\mu\in G} \mathcal{H}_{s,M}(\mu) - e^{\star}_{\varepsilon} \Big). \tag{A.69}$$

Combining Eqs. (A.65) and (A.69) and applying Definition (5) establishes that $(\mu_N)_{N\geq 1}$ satisfies an LDP under $\mathbb{P}^{\varepsilon}_{N,\beta_N}$ with speed β_N and good rate function

$$\mathcal{J}^{\varepsilon}_s(\mu) \stackrel{\text{def}}{=} \mathcal{H}^{\varepsilon}_s(\mu) - e^{\star}_{\varepsilon} = \mathcal{H}^{\varepsilon}_s(\mu) - \inf_{\nu\in\mathcal{P}(\Omega)} \mathcal{H}^{\varepsilon}_s(\nu).$$

Finally, let $\varepsilon \downarrow 0$. Since $g^{\varepsilon}_s \uparrow g_s$, we have $\mathcal{H}^{\varepsilon}_s(\mu) \uparrow \mathcal{H}_s(\mu)$ pointwise (possibly $+\infty$), and

$$e^{\star}_{\varepsilon} = \inf_{\mu} \mathcal{H}^{\varepsilon}_s(\mu) \uparrow \inf_{\mu} \mathcal{H}_s(\mu) \stackrel{\text{def}}{=} e^{\star}.$$

Thus $\mathcal{J}^{\varepsilon}_s(\mu) \uparrow \mathcal{J}_s(\mu)$ pointwise, where

$$\mathcal{J}_s(\mu) \stackrel{\text{def}}{=} \mathcal{H}_s(\mu) - e^{\star} = \mathcal{H}_s(\mu) - \inf_{\nu\in\mathcal{P}(\Omega)} \mathcal{H}_s(\nu),$$

which is the claimed rate function in the regime $\beta_N/N \to \infty$.

A.4.5. GOODNESS OF RATE FUNCTION.

In the first regime, $\mathrm{Ent}(\cdot \mid \ell)$ has compact level sets (Sanov rate is good), and $\mathcal{E}_s$ is lower semicontinuous, so $\mathcal{E}_s + \mathrm{Ent}(\cdot \mid \ell)$ has compact sublevel sets, hence $\mathcal{J}_s$ is good. In the second regime, $\mathcal{E}_s$ is lower semicontinuous and the underlying space is bounded, so the sublevel sets of $\mathcal{E}_s$ are tight; together with lower semicontinuity this yields goodness.