Markov-k Transformer LPE Report (k=1)

Automated run

February 19, 2026

Artifacts Used

- Transformer artifacts: `artifacts/markov_k1_gaplt1_highcap_run2`
- Report figures and diagnostics: `artifacts/markov_k1_gaplt1_highcap_report/figures`

1. Model Architecture

Field	Value
Transformer type	Decoder-only, causal self-attention
Positional encoding	Learned absolute positional embeddings
Markov order	$k = 1$
Layers (L)	6
Model width (d_{model})	128
Heads (H)	8
MLP width (d_{mlp})	512
Pre-norm	True
Trainable parameters	1,450,498

Table 1: k=1 transformer architecture used for this report.

2. Training Details

Field	Value
Training objective	Autoregressive cross-entropy on next-token prediction
Sequence length	2000
Batch size	8
Gradient accumulation	8
Optimizer	AdamW
Learning rate	3×10^{-4} (cosine decay to 2×10^{-6})
Weight decay	0.001
Warmup schedule	Linear warmup for 200 steps
Gradient clipping	1.0
Total steps run	3000 (early-stop target reached)
Early-stop target	Step-1 gap $\leq 0.99\%$

Table 2: Training setup for the high-capacity k=1 run.

Data generation process (explicit Markov parameterization):

- General k-Markov parameterization:

$$\theta_s \equiv P(x_t = 1 \mid x_{t-k:t-1} = s), \qquad s \in \{0,1\}^k.$$

- Prior over parameters: independent Beta per state

$$\theta_s \sim \mathrm{Beta}(1,1) \ \text{ independently for all } s \in \{0,1\}^k.$$

- Sequence generation for one training example of length $T = 2000$:

$$x_1, \ldots, x_k \overset{\text{i.i.d.}}{\sim} \mathrm{Bernoulli}(0.5),$$

$$x_t \mid x_{t-k:t-1} \sim \mathrm{Bernoulli}\big(\theta_{x_{t-k:t-1}}\big), \qquad t \geq k+1.$$

- For the present report ($k = 1$), this reduces to

$$\theta_0 = P(x_t = 1 \mid x_{t-1} = 0), \qquad \theta_1 = P(x_t = 1 \mid x_{t-1} = 1),$$

$$\theta_0, \theta_1 \overset{\text{i.i.d.}}{\sim} \mathrm{Beta}(1,1), \quad x_1 \sim \mathrm{Bernoulli}(0.5), \quad x_t \mid x_{t-1} \sim \mathrm{Bernoulli}(\theta_{x_{t-1}}) \ (t \geq 2).$$

- Equivalent (p, q) form for $k = 1$: $p = P(1 \mid 0) = \theta_0$, $q = P(0 \mid 1) = 1 - \theta_1$.

3. Training Curve

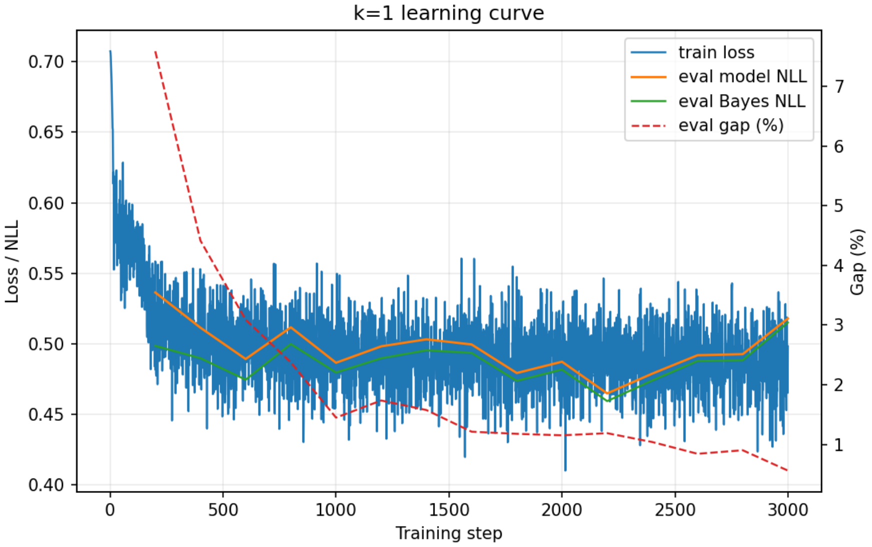

Figure 1: Training curve for k=1. Training loss is logged every gradient step. Evaluation curves are drawn only at true eval checkpoints (every 200 steps) and linearly connected between checkpoints.

4. Final Loss vs Bayes Predictive

Held-out evaluation uses sequences from the same Beta-Bernoulli Markov-1 process.

For a prefix $x_{1:t-1}$ with transition counts $n_{00}, n_{01}, n_{10}, n_{11}$, the Bayes predictive is

$$P(x_t = 1 \mid x_{1:t-1}) = \begin{cases} \dfrac{1+n_{01}}{2+n_{00}+n_{01}}, & x_{t-1} = 0, \\ \dfrac{1+n_{11}}{2+n_{10}+n_{11}}, & x_{t-1} = 1. \end{cases}$$

The per-token Bayes NLL is

$$\ell_t^{\text{Bayes}} = -\big[x_t \log q_t + (1-x_t)\log(1-q_t)\big],$$

where $q_t = P(x_t = 1 \mid x_{1:t-1})$.

Metric	Value	Notes
Model NLL	0.467672	Per-token negative log-likelihood
Bayes NLL	0.463674	Exact Bayes predictive on held-out samples
Gap	0.86%	(model − Bayes)/Bayes

Table 3: Model predictive quality against Bayes baseline (k=1).

To compare next-token probabilities directly, we generated 200 mixed-length contexts:

- Sample context length uniformly from $[64, 2031]$.

- Sample one latent Markov process from the priors above and draw context bits.
- Compute Bayes and model $P(x_{n+1} = 1 \mid x_{1:n})$.

Summary: Pearson $r = 0.9918$, MAE $= 0.02542$.

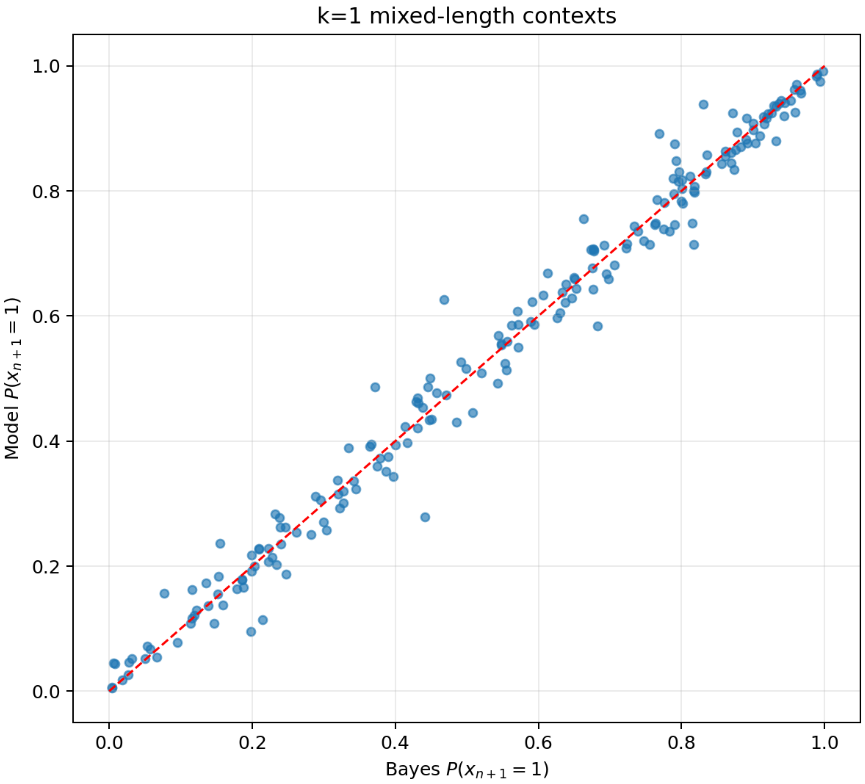

Figure 2: Model vs Bayes next-token probability on mixed-length contexts for k=1.

5. Posterior-Sample Quality

Diagnostics used 30 trials, posterior samples per trial=200, rollout length=1000. For this section, each trial corresponds to one generated input context. Contexts were generated as:

- Sample latent transitions independently: $\theta_0, \theta_1 \sim \text{Beta}(1, 1)$.
- Sample one context of length 1000 from the Markov-1 process:

$$x_1 \sim \text{Bernoulli}(0.5), \qquad x_t \mid x_{t-1} \sim \text{Bernoulli}(\theta_{x_{t-1}}),\ t \geq 2.$$

- Compute exact Bayes posterior marginals from context transition counts:

$$\theta_0 \mid x_{1:n} \sim \text{Beta}(n_{01} + 1, n_{00} + 1), \qquad \theta_1 \mid x_{1:n} \sim \text{Beta}(n_{11} + 1, n_{10} + 1).$$

- Estimate model-implied posterior by 200 rollouts of length 1000 per context; each rollout yields one $(\hat{\theta}_0, \hat{\theta}_1)$ from continuation transition frequencies.

Metric	Mean	Median	p90	p95	Max	Perfect-match expected mean
Wasserstein-1	0.0349	0.0213	0.0908	0.0978	0.1928	0.0020
KS(CDF)	0.4354	0.3804	0.7507	0.8755	0.9797	0.0570
CvM-int	0.01830	0.00609	0.04140	0.06649	0.15606	0.00006
PIT-KS	0.4385	0.3847	0.7542	0.8773	0.9805	0.0604
PIT-CvM	22.2617	16.0544	50.4823	59.4791	65.9362	0.1648
Quantile RMSE	0.0353	0.0213	0.0905	0.0976	0.1925	0.0025
Coverage MAE	0.2895	0.2172	0.6036	0.7007	0.7813	0.0200

Table 4: Summary of posterior-distribution similarity metrics across 30 Markov-1 contexts, with finite-sample perfect-match baselines.

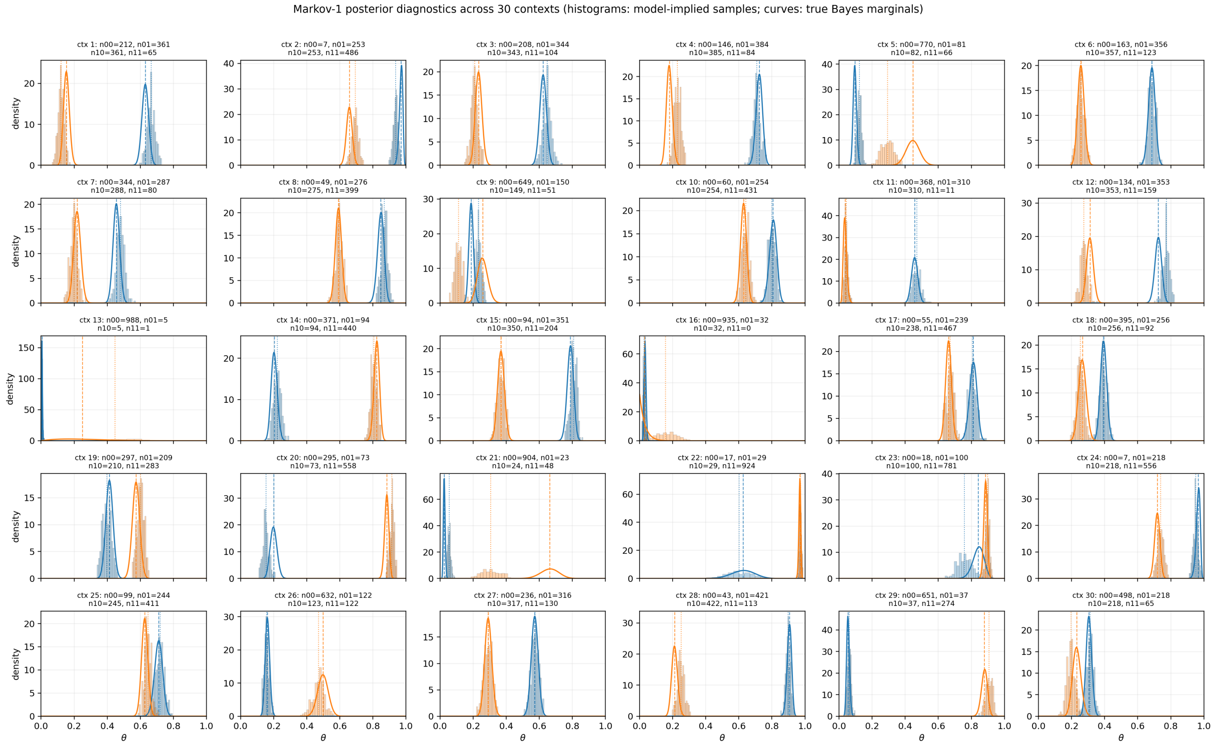

Figure 3: Posterior-shape diagnostics for 30 contexts (5x6 grid). In each panel, histograms are model-implied samples for θ_0, θ_1; curves are true Bayes posterior marginals.

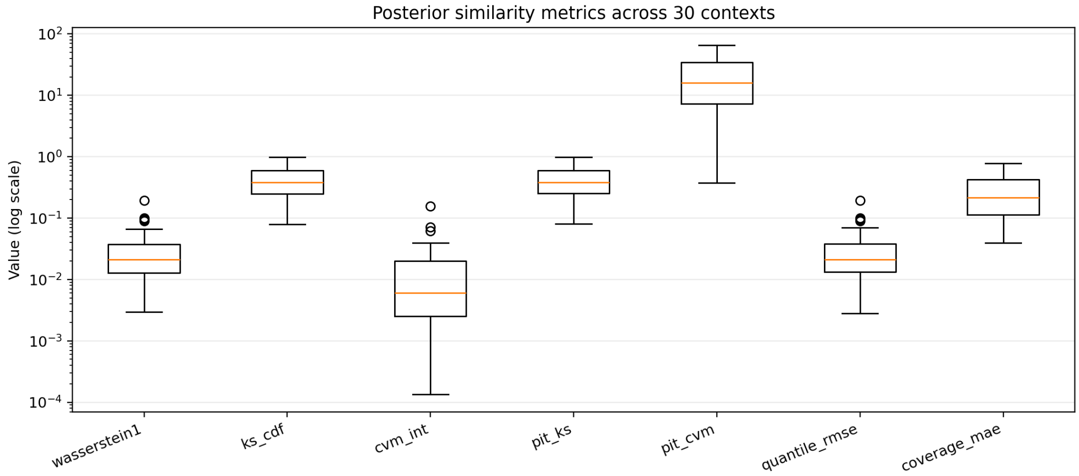

Figure 4: Boxplots of per-context posterior similarity metrics (30 values per metric, log scale).

For each metric, the reported context-level value is computed per state (state 0 and state 1) and then averaged, so this section keeps the same metric family and table structure as the Bernoulli report while adapting to the 2-parameter posterior.

Full per-context and summary metrics are available in: `artifacts/markov_k1_gaplt1_highcap_report/figure` and `artifacts/markov_k1_gaplt1_highcap_report/figures/posterior_context_metrics_summary_5x6_k1.cs`

Overall, posterior-shape match improved relative to earlier runs but remains context-dependent: central tendencies are materially above finite-sample perfect-match baselines, and tail metrics (especially PIT-CvM and high-quantile KS/CvM) are still large on hard contexts.

6. LPE Estimation Quality and Naive MC Baseline

For this k=1 LPE rerun (Step 3 only; same checkpoint as Sections 1–5):

- Number of contexts: 8 (all length 1000).
- Target string length: 100.
- Target mode: `balanced` (de Bruijn-based construction).
- The same fixed target string is used for all 8 contexts in this run.
- Posterior samples per context: $M = 200$; rollout length $L = 1000$.

A binary de Bruijn cycle $B(2, n)$ is a cyclic bit string of length 2^n such that, when read with wrap-around, every length-n binary substring appears exactly once. For example, $B(2, 2)$ can be written as `0011`: its cyclic length-2 windows are `00`, `01`, `11`, `10`, i.e., all four 2-bit patterns exactly once.

The de Bruijn target is generated exactly as follows (matching the code path in `make_target_bits`):

1. Build the binary de Bruijn cycle $B(2, k+1)$. For $k = 1$, this is order 2 and one representative cycle is $c =$ `0011` (up to rotation).
2. Sample a random cyclic offset $r \in \{0, \ldots, |c| - 1\}$.

3. Rotate the cycle by r, then repeat this rotated cycle and truncate to length $m = 100$:

$$y_t = c_{(t+r) \bmod |c|}, \quad t = 1, \ldots, m.$$

For $k = 1$, this gives an exactly balanced target in this run: $m = 100$ is a multiple of $|c| = 4$, so the target has 50 zeros and 50 ones, and each 1-step state (0 or 1) is followed equally often by 0 and 1 over the target.

True event probability is computed analytically under Bayes predictive using Beta-function ratios by state:

$$P(y_{1:m} \mid x_{1:n}) = \prod_{s\in\{0,1\}} \frac{B(\alpha + n_{s,1} + m_{s,1},\ \beta + n_{s,0} + m_{s,0})}{B(\alpha + n_{s,1},\ \beta + n_{s,0})}, \quad \alpha = \beta = 1.$$

Here $x_{1:n}$ is the observed context (length n), $y_{1:m}$ is the fixed target string (length m), and s indexes the 1-step Markov state ($s \in \{0, 1\}$ for $k = 1$; in general $s \in \{0, 1\}^k$). The counts $n_{s,1}$ and $n_{s,0}$ are the number of transitions in the context from state s to next bit 1 and 0, respectively. The counts $m_{s,1}$ and $m_{s,0}$ are the corresponding transition counts contributed by the target when it is appended after the context (including the bridge transition from the context into the target).

The rollout-posterior estimator is

$$\widehat{P}_{\text{post}}(y_{1:m} \mid x_{1:n}) = \frac{1}{M} \sum_{j=1}^{M} f(\hat{\theta}^{(j)}),$$

where

$$f(\theta) \equiv P_\theta(y_{1:m} \mid x_{1:n}) = \prod_{t=1}^{m} \theta_{s_t}^{y_t} \left(1 - \theta_{s_t}\right)^{1-y_t},$$

and s_t is the Markov state immediately before target bit y_t in the concatenated history $x_{1:n}, y_{1:t-1}$. For $k = 1$, $\theta = (\theta_0, \theta_1)$ with $\theta_s = P(1 \mid s)$. Each $\hat{\theta}^{(j)}$ is obtained from one rollout's transition frequencies.

Equal-compute naive MC uses

$$R_{\text{eq}} = \left\lfloor \frac{ML}{m} \right\rfloor = \left\lfloor \frac{200 \times 1000}{100} \right\rfloor = 2000,$$

and

$$\text{relative SE}_{\text{naive}} = \sqrt{\frac{1-P}{R_{\text{eq}} P}}, \qquad P(\text{zero hits}) = (1-P)^{R_{\text{eq}}}.$$

Statistic	Posterior method $\lvert$rel err$\rvert$	Naive MC expected rel SE	Naive MC P(zero hits)
p50	1.00×10^{2}	3.27×10^{10}	1.000000
p90	2.13×10^{47}	1.76×10^{45}	1.000000
p95	4.62×10^{47}	3.82×10^{45}	1.000000

Table 5: k=1 posterior estimator error vs expected naive-MC error under matched token budget.

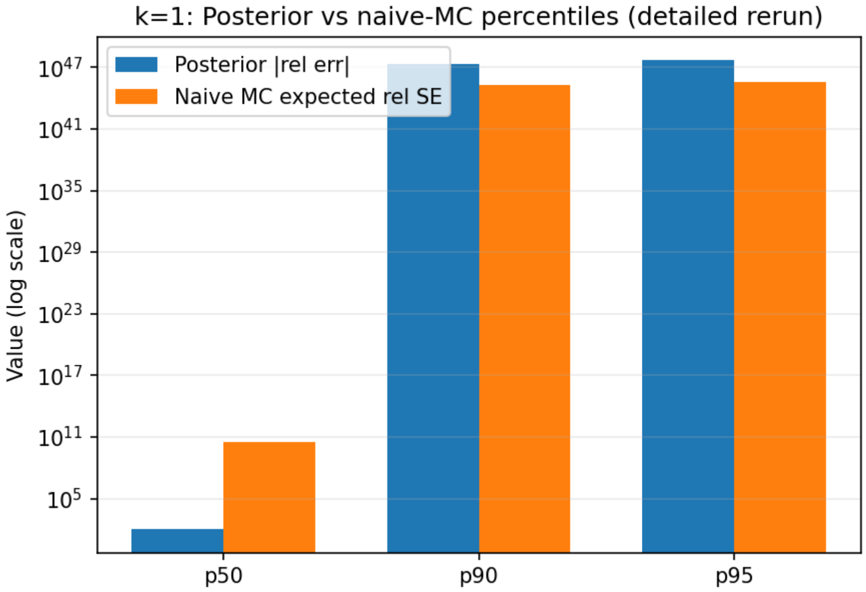

Figure 5: Percentile-level comparison of posterior method error and naive-MC expected error (k=1, log scale).

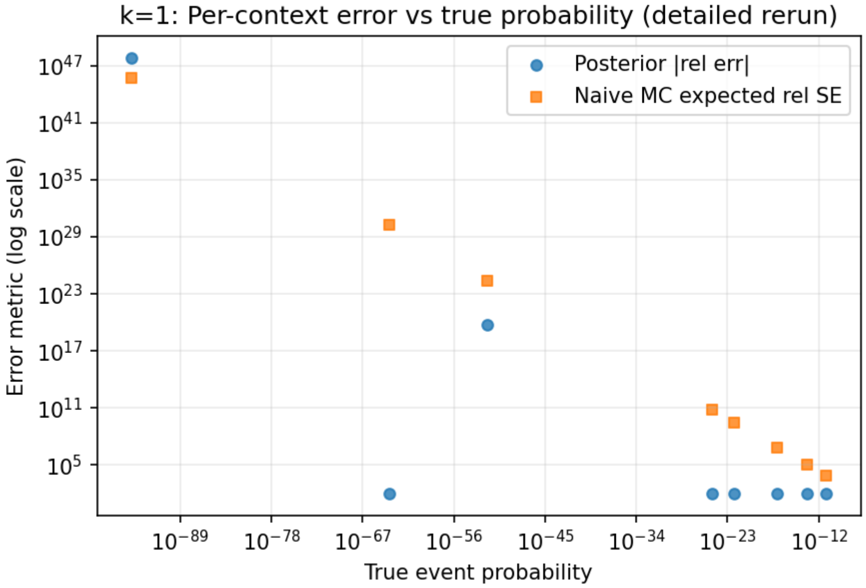

Figure 6: Per-context relative error versus true event probability (k=1, log-log).

Context	True prob	Predicted prob	True posterior mean	True posterior sd	Sampled posterior mean	Sampled posterior sd
0	1.288e-25	1.166e-32	(0.465,0.686)	(0.025,0.019)	(0.453,0.699)	(0.027,0.023)
1	1.568e-64	1.558e-67	(0.958,0.036)	(0.009,0.008)	(0.956,0.035)	(0.010,0.009)
2	5.420e-23	6.729e-33	(0.522,0.693)	(0.026,0.018)	(0.503,0.724)	(0.030,0.022)
3	9.091e-18	1.362e-53	(0.450,0.982)	(0.078,0.004)	(0.513,0.980)	(0.100,0.006)
4	7.080e-12	3.757e-41	(0.846,0.741)	(0.024,0.016)	(0.889,0.736)	(0.022,0.020)
5	4.142e-14	1.553e-48	(0.591,0.957)	(0.060,0.007)	(0.538,0.961)	(0.066,0.008)
6	1.449e-95	1.030e-49	(0.052,0.091)	(0.007,0.038)	(0.047,0.159)	(0.009,0.055)
7	1.088e-52	5.626e-35	(0.215,0.393)	(0.015,0.030)	(0.189,0.443)	(0.020,0.038)

Table 6: Per-context posterior diagnostics for the Step-3 rerun. Posterior mean/sd entries are shown as (θ_0, θ_1), where $\theta_0 = P(1 \mid 0)$ and $\theta_1 = P(1 \mid 1)$.

In this rerun, the posterior method improves the median error substantially relative to equal-compute naive MC (p50: 10^2 vs 3.27×10^{10}), but upper-tail errors remain very large and exceed the naive baseline at p90/p95. This indicates that the posterior-based estimator helps on easier contexts while still breaking down on the hardest rare-event contexts.

The table supports a concrete mechanism for the large LPE failures: on hard contexts, the model-implied posterior can be materially shifted from the true posterior. For example, context 6 has true posterior mean $(0.052, 0.091)$ but sampled posterior mean $(0.047, 0.159)$, i.e., a large upward shift in θ_1. Context 7 shows a similar θ_1 upward shift $(0.393 \to 0.443)$. These posterior mismatches are enough to strongly distort the estimated target probability, especially when the true event probability is already extremely small.