Analytical Correction for Subsampling Bias in Drifting Models

Jiaru Zhang, Zeyun Deng, Juanwu Lu, Ziran Wang*, Ruqi Zhang*
Purdue University
Correspondence to: jiaru@purdue.edu

Abstract

Drifting models are capable one-step generative models trained to follow a drifting field. The field combines attractive and repulsive softmax-weighted centroids over the data and current-generator distributions. In practice, only a minibatch of n samples from each distribution is available, and each centroid is approximated by an empirical estimate. In this paper, we begin by showing that the minibatch centroid is in general a *biased* estimator of the target centroid, with a pointwise $O(1/n)$ bias arising from softmax self-normalization. Correcting this bias requires the expectation over the full distribution, which is intractable. We instead approximate the leading bias term from in-batch statistics and propose *Analytical Bias Correction* (ABC), a closed-form plug-in adjustment. We prove that ABC reduces the bias from $O(1/n)$ to $O(1/n^2)$, introduces no first-order increase in total variance, and preserves convex-hull containment of the corrected centroid. In practice, ABC requires only two additional lines of code and has negligible wall-time overhead under compiled execution. Toy experiments confirm the theoretical $O(1/n)$ and $O(1/n^2)$ scaling. On CIFAR-10, ABC reduces FID and trains faster, with the largest gains at small n, where the bias is most significant.

1 Introduction

Drifting models [7] are capable one-step generative models trained to follow a drifting field. Each generator sample defines a query point at which the field combines attraction toward a softmax-weighted centroid of the data distribution with repulsion from that of the current generator distribution. Despite being introduced only recently, they have already been extended to molecular sampling, medical image synthesis, and control [8, 14, 20], pointing to broad application potential. Unlike diffusion models or flow matching, whose target field is derived from a predefined forward process, drifting models construct this target field directly from data and generator distributions. Since computing this field over the full distributions is prohibitive, the standard practice is to substitute a minibatch of n samples from each distribution.

As already shown by Deng et al. [7], small n degrades generation quality sharply. What is less appreciated is *why*. In this paper, we start with an exact analysis of the $n{=}1$ case where the minibatch centroid reduces to a single sample, and find that its expectation reduces to the global mean rather than the softmax-weighted local average that defines the target. For general n, we derive the explicit leading-order $O(1/n)$ bias of the softmax-weighted centroid at any query and show that the bias systematically *contracts* the centroid toward a less localized average and away from the nearest data mode. The bias is therefore not stochastic noise but a structural property of the minibatch estimator that should be *corrected*.

Figure 1 visualizes this on a 2D toy example with three Gaussian clusters and a query (star) closer to the left cluster than the other two. The target centroid (**green diamond**) is a softmax-weighted

*Equal advising.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

average pulled toward the nearest cluster. A minibatch of $n{=}4$ samples (black rings) produces a biased centroid (**red circle**) shifted back toward the global mean of the three clusters, with the residual pointing *away* from the nearest cluster. This is the centroid-contraction bias we characterize.

Exact correction would require expectations over the full reference distributions, which are intractable. We instead approximate the leading bias term from *in-batch* statistics and derive **Analytical Bias Correction (ABC)**, a closed-form plug-in adjustment that reuses quantities already computed for the softmax. Although ABC is itself estimated from in-batch statistics, we prove that it reduces the leading-order bias from $O(1/n)$ to $O(1/n^2)$, introduces no first-order increase in total variance, and preserves convex-hull containment of the corrected centroid. Figure 1 shows the corrected centroid (**blue square**) sitting visibly closer to the target centroid than the biased one. In practice, the correction adds two lines of code and has negligible wall-time overhead under compiled execution. Toy experiments confirm the predicted $O(1/n)$ and $O(1/n^2)$ scaling directly, and on CIFAR-10, ABC yields lower FID and reaches given FID targets with fewer training samples, with the largest gains at small n where the bias is most severe. The code is available at `https://github.com/lan-qing/DriftingABC`.

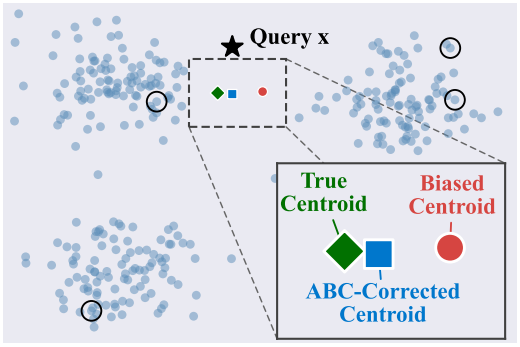


Figure 1: 2D toy with three Gaussian clusters. From a minibatch of $n{=}4$ (black rings), the biased centroid (**red circle**) drifts away from the target centroid (**green diamond**), and ABC pulls it back (**blue square**).

In summary, our contributions are:

1. We show that, at any fixed query $\boldsymbol{x}$, the minibatch centroid in drifting models is a biased estimator of the target centroid, with a pointwise $O(1/n)$ bias arising from softmax self-normalization.
2. We derive ABC by approximating the leading-order bias from in-batch statistics, and prove it achieves $O(1/n^2)$ residual bias, convex-hull containment, and no first-order increase in total variance.
3. Toy experiments confirm the theoretical $O(1/n)$ and $O(1/n^2)$ scaling directly. On CIFAR-10, ABC lowers FID and accelerates convergence, with the largest gains at small n.

1.1 Related Work

Drifting models. Drifting models were introduced by Deng et al. [7] as one-step generators trained with kernel-based attraction-repulsion fields. Since then, a number of concurrent works have developed theoretical views and alternative formulations [4, 9, 10, 18], and extended drifting models beyond image generation to molecular sampling, medical image synthesis, and control [8, 14, 20]. Among them, the concurrent work of Lai et al. [18] establishes an equivalence between drifting and score-matching under Gaussian/Laplace kernels, and in the course of this analysis, they also remark that finite-sample mini-batch drift estimation with normalized kernel weights forms a ratio-type Monte Carlo estimator and may therefore exhibit variance and bias. However, their work focuses on the population-level connection to score matching, not on the finite-sample estimator itself. Complementary to that work, we make the bias precise by characterizing it as a generic $O(1/n)$ centroid contraction, and derive an analytical correction with $O(1/n^2)$ residual bias.

Self-normalization, minibatching, and generative transport. Bias in ratio estimators and self-normalized Monte Carlo estimators has long been understood. Classical analyses characterize the leading $O(1/n)$ term, and bias-reduction schemes such as the jackknife or Bias Reduced Self-Normalized Importance Sampling (BR-SNIS) can push the residual to $O(1/n^2)$ [1, 5, 6, 22, 23]. However, these schemes typically incur extra kernel launches and $O(nD)$ leave-one-out bookkeeping per evaluation or require auxiliary samples, which is prohibitive in drifting models where the centroid must be recomputed at every training step. The jackknife is additionally unstable when one sample dominates the softmax, because removing that sample produces a leave-one-out centroid far from the

rest and the jackknife's bias-correction step amplifies this outlier. Our contribution is a closed-form correction that integrates naturally with the existing softmax computation. Related effects also arise when machine learning objectives approximate a full normalizer with sampled or truncated support, such as sampled softmax and efficient attention [2, 27]. Our setting is different from both lines of work and from flow matching [19], as drifting trains against a softmax-weighted centroid over the *entire* reference distribution, so minibatching produces a self-normalized target estimator rather than an unbiased sample average.

2 Preliminaries

The drifting model. The drifting model [7] generates samples by training-time evolution of a pushforward distribution. At each step, a *drifting field* $\boldsymbol{V}_{p,q}(\boldsymbol{x})$ prescribes the target direction at each generated sample $\boldsymbol{x}$ for training the model, depending on both the data distribution p and the current generator distribution q. The field is defined as: $\boldsymbol{V}_{p,q}(\boldsymbol{x}) = \boldsymbol{V}^+_p(\boldsymbol{x}) - \boldsymbol{V}^-_q(\boldsymbol{x})$, where the *positive* (attraction) and *negative* (repulsion) components are softmax-weighted mean-shift vectors:

$$\boldsymbol{V}^+_p(\boldsymbol{x}) := \frac{1}{Z_p(\boldsymbol{x})}\,\mathbb{E}_{\boldsymbol{y}^+\sim p}\big[k(\boldsymbol{x},\boldsymbol{y}^+)(\boldsymbol{y}^+ - \boldsymbol{x})\big], \qquad \boldsymbol{V}^-_q(\boldsymbol{x}) := \frac{1}{Z_q(\boldsymbol{x})}\,\mathbb{E}_{\boldsymbol{y}^-\sim q}\big[k(\boldsymbol{x},\boldsymbol{y}^-)(\boldsymbol{y}^- - \boldsymbol{x})\big], \tag{1}$$

with normalization factors $Z_p(\boldsymbol{x}) = \mathbb{E}_p[k(\boldsymbol{x},\boldsymbol{y}^+)]$ and $Z_q(\boldsymbol{x}) = \mathbb{E}_q[k(\boldsymbol{x},\boldsymbol{y}^-)]$. The kernel $k(\cdot,\cdot)$ can be any positive similarity function, and we adopt $k(\boldsymbol{x},\boldsymbol{y}) = \exp(-\|\boldsymbol{x}-\boldsymbol{y}\|_2/\tau)$ by default with temperature τ following Deng et al. [7]. Intuitively, $\boldsymbol{V}^+_p$ attracts $\boldsymbol{x}$ toward nearby data samples and $\boldsymbol{V}^-_q$ repels it from other generated samples, and the combined field drives the generator distribution toward the data distribution.

Minibatch approximation. In practice, the expectations over p and q in Equation (1) are intractable. Instead, at each training step, n reference samples $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ are drawn as a minibatch, and the positive field $\boldsymbol{V}^+_p(\boldsymbol{x})$ is replaced by

$$\sum_{i=1}^{n} \alpha_i\,(\boldsymbol{y}_i - \boldsymbol{x}), \qquad \text{where} \quad \alpha_i = \frac{k(\boldsymbol{x},\boldsymbol{y}_i)}{\sum_{j=1}^{n} k(\boldsymbol{x},\boldsymbol{y}_j)}, \tag{2}$$

which is a softmax-weighted combination over the minibatch. The negative field is approximated similarly from other concurrently generated samples, excluding the query itself.

3 The Bias Problem

In this section, we first show that the minibatch estimator is biased for n=1 with an exact result. As the exact bias admits no closed form for general n, we then analyze the zero-bias conditions and quantify the leading-order bias. Note that the positive component is a softmax-weighted mean-shift over the data distribution p and the negative is its structural counterpart over the generator distribution q, and both have the same ratio-of-averages form. For simplicity, we focus on the positive side in what follows and omit the p subscript on $\mathbb{E}[\cdot]$. Similar statements apply to the negative side by replacing p with q throughout. For ease of reference, we also present a table of notations in Appendix A.

Centroid reformulation. To simplify the analysis, we first rewrite the formulation in centroid form. Concretely, we rewrite the mean-shift vector in Equation (1) as $\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{T}^*(\boldsymbol{x}) - \boldsymbol{x}$, where

$$\boldsymbol{T}^*(\boldsymbol{x}) \;:=\; \frac{\mathbb{E}\big[k(\boldsymbol{x},\boldsymbol{y})\,\boldsymbol{y}\big]}{\mathbb{E}\big[k(\boldsymbol{x},\boldsymbol{y})\big]} \tag{3}$$

is the *target centroid*, i.e., the softmax-weighted average of the reference distribution. The minibatch approximation in Equation (2) estimates $\boldsymbol{T}^*$ by

$$\boldsymbol{T}_n(\boldsymbol{x}) \;:=\; \sum_{i=1}^{n} \alpha_i\,\boldsymbol{y}_i, \qquad \text{where} \quad \alpha_i = \frac{k(\boldsymbol{x},\boldsymbol{y}_i)}{\sum_{j=1}^{n} k(\boldsymbol{x},\boldsymbol{y}_j)}. \tag{4}$$

Since $\boldsymbol{x}$ is fixed, it suffices to analyze the bias of $\boldsymbol{T}_n$ relative to $\boldsymbol{T}^*$.

The bias in the single-sample case $n = 1$. When n=1, a single sample is drawn from the distribution and used directly as the centroid estimate, i.e., $\alpha_1 = 1$ and $\boldsymbol{T}_1 = \boldsymbol{y}$, and the softmax-weighted

averaging is lost entirely. The expected estimate therefore reduces to the *global mean* $\mathbb{E}[\boldsymbol{y}]$, whereas the target $\boldsymbol{T}^*$ is a *softmax-weighted* average that puts more mass on points close to $\boldsymbol{x}$. The bias is exactly the gap between the two. To state it precisely, write $w = k(\boldsymbol{x}, \boldsymbol{y})$ for the kernel weight, and we have:

Theorem 3.1 (Bias for n=1)**.** *For a single sample $\boldsymbol{y} \sim p$ with $w = k(\boldsymbol{x}, \boldsymbol{y}) \geq 0$, $\Pr(w > 0) = 1$ (so $\boldsymbol{T}_1$ is well defined), $\mathbb{E}[w] \in (0, \infty)$, $\mathbb{E}[\|\boldsymbol{y}\|] < \infty$, and $\mathbb{E}[\|w\boldsymbol{y}\|] < \infty$:*

$$\mathbb{E}[\boldsymbol{T}_1] - \boldsymbol{T}^* \;=\; -\frac{\mathrm{Cov}(w,\, \boldsymbol{y})}{\mathbb{E}[w]}, \tag{5}$$

where $\mathrm{Cov}(w, \boldsymbol{y}) := \mathbb{E}[(w - \mathbb{E}w)(\boldsymbol{y} - \mathbb{E}\boldsymbol{y})] \in \mathbb{R}^D$ is the componentwise covariance between the scalar w and the vector $\boldsymbol{y}$.

Proof. For n=1 the centroid reduces to the sample itself, so $\mathbb{E}[\boldsymbol{T}_1] = \mathbb{E}[\boldsymbol{y}]$. The target centroid is $\boldsymbol{T}^* = \mathbb{E}[w\boldsymbol{y}]/\mathbb{E}[w]$. Their difference is

$$\mathbb{E}[\boldsymbol{y}] - \frac{\mathbb{E}[w\boldsymbol{y}]}{\mathbb{E}[w]} \;=\; \frac{\mathbb{E}[w]\,\mathbb{E}[\boldsymbol{y}] - \mathbb{E}[w\boldsymbol{y}]}{\mathbb{E}[w]} \;=\; -\frac{\mathrm{Cov}(w, \boldsymbol{y})}{\mathbb{E}[w]}. \qquad \square$$

The bias is governed by $\mathrm{Cov}(w, \boldsymbol{y})$, the covariance between kernel weight and sample position. Since points closer to $\boldsymbol{x}$ receive larger w, this covariance is generically nonzero, and the negative sign in Equation (5) causes the estimator's expectation to shift away from the softmax-weighted target toward a flatter average.

Conditions for the bias to vanish in general n. The n=1 result admits an exact formula because the softmax weight trivially equals one. For general n, the weights $\alpha_i = k(\boldsymbol{x}, \boldsymbol{y}_i)/\sum_j k(\boldsymbol{x}, \boldsymbol{y}_j)$ couple all samples through a shared denominator, and $\mathbb{E}[\boldsymbol{T}_n]$ no longer has a closed-form expression. Nonetheless, we can identify two natural sufficient conditions under which the bias vanishes:

Theorem 3.2 (Zero-bias conditions)**.** *Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \overset{\text{i.i.d.}}{\sim} p$ with $w = k(\boldsymbol{x}, \boldsymbol{y}) \geq 0$, $\mathbb{E}[w] \in (0, \infty)$, $\mathbb{E}[\|\boldsymbol{y}\|] < \infty$, $\mathbb{E}[\|w\boldsymbol{y}\|] < \infty$, and $\Pr(w > 0) = 1$ (so $\sum_i w_i > 0$ a.s. and $\boldsymbol{T}_n$ is well defined). $\mathbb{E}[\boldsymbol{T}_n] = \boldsymbol{T}^*$ if either:*

(i) Constant weight*: $w = k(\boldsymbol{x}, \boldsymbol{y})$ is constant a.s. (e.g., all points equidistant from $\boldsymbol{x}$ for a radial kernel); or*

(ii) Kernel-symmetric data*: p is symmetric about $\boldsymbol{x}$ and the kernel is reflection-invariant ($k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x}, 2\boldsymbol{x} - \boldsymbol{y})$; holds for any radial kernel).*

Proof. (i). Constant w gives $\alpha_i = 1/n$, so $\boldsymbol{T}_n = n^{-1}\sum_i \boldsymbol{y}_i$ and $\mathbb{E}[\boldsymbol{T}_n] = \mathbb{E}[\boldsymbol{y}] = \boldsymbol{T}^*$.

(ii). First, we show $\boldsymbol{T}^* = \boldsymbol{x}$. By symmetry, $\mathbb{E}[w(\boldsymbol{y} - \boldsymbol{x})] = -\mathbb{E}[w(\boldsymbol{y} - \boldsymbol{x})]$ (the reflection $\boldsymbol{y} \mapsto 2\boldsymbol{x} - \boldsymbol{y}$ preserves both p and w), so $\mathbb{E}[w(\boldsymbol{y} - \boldsymbol{x})] = \boldsymbol{0}$ and $\boldsymbol{T}^* = \mathbb{E}[w\boldsymbol{y}]/\mathbb{E}[w] = \boldsymbol{x}$. For $\boldsymbol{T}_n$, reflecting the minibatch does not change its probability or the weights, but the centroid becomes $2\boldsymbol{x} - \boldsymbol{T}_n$. Therefore $\mathbb{E}[\boldsymbol{T}_n] = \mathbb{E}[2\boldsymbol{x} - \boldsymbol{T}_n] = 2\boldsymbol{x} - \mathbb{E}[\boldsymbol{T}_n]$, giving $\mathbb{E}[\boldsymbol{T}_n] = \boldsymbol{x} = \boldsymbol{T}^*$. $\square$

Both conditions are restrictive. Condition (i) requires all points to be equidistant from the query, and (ii) requires the data to be symmetric about it. Outside these special cases, how large is the bias? The next result answers this for general n.

The magnitude of the bias in general n. Although an exact closed form is unavailable, we can still derive the leading order of the bias, stated in the following theorem.

Theorem 3.3 (Leading-order bias)**.** *Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \overset{\text{i.i.d.}}{\sim} p$ with $w = k(\boldsymbol{x}, \boldsymbol{y}) \in [0, M]$ and $\|\boldsymbol{y}\| \leq C$ almost surely for some constants $M, C > 0$, and $\Pr(w > 0) = 1$ (so $\sum_i w_i > 0$ a.s. and $\boldsymbol{T}_n$ is well defined). For fixed $\boldsymbol{x}$ with $\mathbb{E}[w] > 0$, the bias of the minibatch centroid satisfies*

$$\mathbb{E}[\boldsymbol{T}_n] - \boldsymbol{T}^* \;=\; -\frac{1}{n}\,\frac{\mathbb{E}[w^2(\boldsymbol{y} - \boldsymbol{T}^*)]}{(\mathbb{E}[w])^2} \;+\; O(n^{-2}) \tag{6}$$

as $n \to \infty$.

The proof views $\boldsymbol{T_n}$ as a ratio of two sample averages, $\boldsymbol{T_n} = \sum_i w_i \boldsymbol{y}_i / \sum_i w_i$. Each sum is individually unbiased, but the expectation of the ratio differs from the ratio of expectations, and the leading-order gap is controlled by the classical delta method for self-normalized estimators [11, 22, 26]. See Appendix B.1 for the full derivation.

The term $\mathbb{E}[w^2(\boldsymbol{y} - \boldsymbol{T^*})]$ is a w^2-*weighted displacement*. Since w^2 amplifies large weights more aggressively than w does, this vector points from $\boldsymbol{T^*}$ toward the more localized w^2-weighted center of mass. The **minus sign** reverses it, so the minibatch estimator is systematically pulled *away* from this local concentration toward a flatter average, a **centroid contraction**. In clustered reference distributions such as the toy in Figure 1, this is visible as a pull from the nearest mode toward the global mean. The $1/n$ factor means the bias shrinks with larger minibatch size but remains nonzero at finite n in general cases.

Remark 3.4 (Assumptions in practice). The boundedness and moment conditions in Theorems 3.1 to 3.3, which are inherited by the ABC results in Section 4, hold naturally in the drifting-model setting for both the positive (data) and negative (current generator) sides. The exponential kernel is strictly positive and bounded, $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|_2/\tau) \in (0, 1]$, so $\Pr(w > 0) = 1$. The reference features $\boldsymbol{y}$ come from a frozen encoder applied to a finite dataset on the positive side and to a batch of generated images on the negative side. In both cases, $\|\boldsymbol{y}\|$ is bounded almost surely, giving finite moments of all orders.

From diagnosis to correction. Given that the bias is $O(1/n)$ with a *closed-form* leading term, a natural plan is to subtract it. Two questions arise immediately. First, the leading term involves full-distribution expectations that are not available at training time, so can we plug in minibatch estimates without reintroducing an $O(1/n)$ error? Second, bias reduction usually comes at the cost of inflated variance, and can we avoid paying that cost? The next section answers both, along with a geometric sanity check on the corrected estimator.

4 Analytical Bias Correction

Having established that the minibatch centroid carries an $O(1/n)$ bias, we now construct a closed-form correction that can be computed from the *same minibatch* at negligible cost. We derive the correction and prove that it reduces the bias to $O(1/n^2)$, introduces no first-order increase in total variance, and preserves convex-hull containment. We briefly note its relationship with temperature tuning.

4.1 Construction

Theorem 3.3 gives the leading-order bias $-\frac{1}{n}\mathbb{E}[w^2(\boldsymbol{y} - \boldsymbol{T^*})]/(\mathbb{E}[w])^2$, but this expression involves full-distribution expectations that are unavailable at training time. To solve this problem, we use the key observation that all quantities can be estimated from the same minibatch used to compute $\boldsymbol{T_n}$. Specifically, we replace $\mathbb{E}[\cdot]$ with $\frac{1}{n}\sum_i$, $\boldsymbol{T^*}$ with $\boldsymbol{T_n}$, and $\mathbb{E}[w]$ with $\frac{1}{n}\sum_i w_i$. Substituting these into the bias formula and simplifying yields the plug-in correction:

$$\Delta_n := -\frac{1}{n}\frac{\frac{1}{n}\sum_i w_i^2 (\boldsymbol{y}_i - \boldsymbol{T_n})}{\left(\frac{1}{n}\sum_i w_i\right)^2} = \frac{\sum_i w_i^2 (\boldsymbol{T_n} - \boldsymbol{y}_i)}{\left(\sum_i w_i\right)^2} = \sum_{i=1}^n \alpha_i^2 (\boldsymbol{T_n} - \boldsymbol{y}_i), \tag{7}$$

where $\alpha_i = w_i / \sum_j w_j$ are the softmax weights as defined in Equation (4). This yields our Analytical Bias Correction (ABC) estimator, which subtracts this correction from the original centroid:

$$\boxed{\boldsymbol{T_n^{\mathrm{ABC}}} := \boldsymbol{T_n} - \Delta_n = \left(1 - \textstyle\sum_i \alpha_i^2\right)\boldsymbol{T_n} + \sum_{i=1}^n \alpha_i^2 \boldsymbol{y}_i.} \tag{8}$$

As a sanity check, when weights are uniform ($\alpha_i = 1/n$), $\sum_i \alpha_i^2 = 1/n$ and $\Delta_n = \mathbf{0}$. This is consistent with Theorem 3.2(i), which shows that constant weights yield an unbiased estimator. We would like to note that the plug-in correction in Equation (8) can also be viewed as a specialization of the classical first-order bias correction for ratio estimators [26].

4.2 Formal Guarantees

Since the correction Δ_n is itself estimated from the same biased minibatch, it is natural to ask whether it truly cancels the bias rather than introducing new errors. The following theorem confirms that ABC reduces the bias by one order of n.

Theorem 4.1 (Bias reduction)**.** *Under the assumptions of Theorem 3.3, the ABC estimator reduces the bias from $O(n^{-1})$ to $O(n^{-2})$:*

$$\mathbb{E}\big[\boldsymbol{T}_n^{\mathbf{ABC}} - \boldsymbol{T}^*\big] = O(n^{-2}). \tag{9}$$

Proof sketch. Decompose Δ_n by inserting $\boldsymbol{T}^*$:

$$\Delta_n \;=\; \underbrace{\sum_i \alpha_i^2\,(\boldsymbol{T}^* - \boldsymbol{y}_i)}_{\text{(A)}} \;+\; \underbrace{\Big(\sum_i \alpha_i^2\Big)(\boldsymbol{T}_n - \boldsymbol{T}^*)}_{\text{(B)}}.$$

Term (A) is the sample analog of the leading bias in Theorem 3.3 and matches it up to $O(n^{-2})$ after a Taylor expansion of the denominator around $\mathbb{E}[w]$. Term (B) has expectation $O(n^{-2})$ because $\mathbb{E}[\sum_i \alpha_i^2] = O(n^{-1})$ from bounded weights and $\mathbb{E}[\boldsymbol{T}_n - \boldsymbol{T}^*] = O(n^{-1})$ from Theorem 3.3, with the residual covariance also $O(n^{-2})$ by Cauchy–Schwarz. Combining, $\mathbb{E}[\Delta_n] = (\mathbb{E}[\boldsymbol{T}_n] - \boldsymbol{T}^*) + O(n^{-2})$, hence $\mathbb{E}[\boldsymbol{T}_n^{\mathbf{ABC}} - \boldsymbol{T}^*] = O(n^{-2})$. See Appendix C.1 for the full proof. □

Beyond bias reduction, two practical concerns remain. First, since the correction subtracts Δ_n from $\boldsymbol{T}_n$, the corrected centroid could in principle extrapolate beyond the minibatch points. Second, reducing bias often increases variance. We address both below.

Proposition 4.2 (Convex hull guarantee)**.** *Under the conditions of Theorem 4.1, $\boldsymbol{T}_n^{\mathbf{ABC}} = \sum_i \gamma_i \boldsymbol{y}_i$ with $\gamma_i = \alpha_i(1 - \sum_j \alpha_j^2) + \alpha_i^2 \geq 0$ and $\sum_i \gamma_i = 1$. Hence $\boldsymbol{T}_n^{\mathbf{ABC}}$ lies in the convex hull of $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$.*

Proof. $\sum_j \alpha_j^2 \leq \sum_j \alpha_j = 1$, so $1 - \sum_j \alpha_j^2 \geq 0$ and $\gamma_i \geq 0$. Summing: $\sum_i \gamma_i = (1 - \sum_j \alpha_j^2) + \sum_j \alpha_j^2 = 1$. □

That is, $\boldsymbol{T}_n^{\mathbf{ABC}}$ is always a weighted average of the minibatch points with non-negative weights, and it never extrapolates beyond the data.

Intuitively, Δ_n is an $O(1/n)$ correction while the fluctuations of $\boldsymbol{T}_n$ are of order $O(1/\sqrt{n})$, so subtracting a term that is smaller than the existing noise by a factor of $\sqrt{n}$ should not change the variance at leading order. The next proposition makes this precise.

Proposition 4.3 (No first-order total-variance increase)**.** *Under the conditions of Theorem 4.1:*

$$\mathbb{E}\big[\|\boldsymbol{T}_n^{\mathbf{ABC}} - \mathbb{E}[\boldsymbol{T}_n^{\mathbf{ABC}}]\|^2\big] \;=\; \mathbb{E}\big[\|\boldsymbol{T}_n - \mathbb{E}[\boldsymbol{T}_n]\|^2\big] + O(n^{-3/2}) \;=\; O(n^{-1}). \tag{10}$$

Proof sketch. The proof decomposes $\boldsymbol{T}_n^{\mathbf{ABC}} = \boldsymbol{T}_n - \Delta_n$ and tracks each piece's contribution to the variance. First, $\boldsymbol{T}_n$ is a smooth ratio of two i.i.d. sample means, so the delta method gives variance $O(n^{-1})$. Second, since $\boldsymbol{T}_n$ is a convex combination of the bounded points, $\|\boldsymbol{y}_i - \boldsymbol{T}_n\| \leq 2C$, giving $\|\Delta_n\|^2 \leq 4C^2(\sum_i \alpha_i^2)^2$. The same bounded-weight argument as in Theorem 4.1, $\sum_i \alpha_i^2 \leq M/\sum_j w_j$, yields $\mathbb{E}[(\sum_i \alpha_i^2)^2] = O(n^{-2})$ and hence $\mathbb{E}\|\Delta_n\|^2 = O(n^{-2})$. The cross term in the variance of $\boldsymbol{T}_n - \Delta_n$ is then $O(n^{-3/2})$ by Cauchy–Schwarz, giving the claimed residual. See Appendix C.2 for the full proof. □

That is, ABC preserves the same leading-order $O(1/n)$ variance as the uncorrected estimator.

4.3 Implementation

ABC is a *drop-in replacement* for the minibatch centroid in Equation (4) that leaves the rest of the drifting-model training pipeline untouched, requiring no auxiliary network and no extra minibatch. Since the drifting field contains both a positive and a negative centroid, each with $O(1/n)$ bias, ABC is applied independently to both. Concretely, given the softmax weights $\texttt{alpha} \in \mathbb{R}^{B\times n}$ and features $\texttt{Y} \in \mathbb{R}^{n\times D}$, where B is the number of queries processed in parallel and D is the feature dimension, the corrected centroid requires only two additional lines in PyTorch:

```
a2 = alpha ** 2
T_abc = (1 - a2.sum(-1, keepdim=True)) * (alpha @ Y) + a2 @ Y
```

The full drifting-model training loop with ABC is summarized in Algorithm 1 of Appendix E.

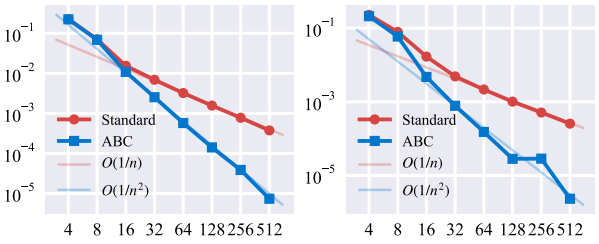
Complexity analysis. We report costs per query. The standard centroid $\boldsymbol{T_n} = \texttt{alpha @ Y}$ costs $O(nD)$. ABC adds an element-wise square of the weights ($O(n)$), a reduction for $\sum_i \alpha_i^2$ ($O(n)$), and one additional matrix multiply $\texttt{a2 @ Y}$ ($O(nD)$). Therefore, the total overhead is $O(nD)$, which is the same order as the baseline.

On the kernel temperature. Since both subsampling bias and a high temperature push the centroid toward the global mean, a natural question is whether tuning the temperature τ alone could substitute for ABC. A first-order analysis in Appendix D shows it cannot, as the bias depends on the *local* weight distribution at each query, whereas τ is a single global scalar, and lowering τ also reduces the effective sample size. A side effect is that the τ inherited from the standard estimator implicitly absorbs part of its $O(1/n)$ bias, so once ABC removes that bias, the optimal τ for the corrected estimator may shift.

5 Experiments

5.1 Toy Experiments

We first verify the bias scaling predicted by Theorem 3.3 on a controlled toy problem before turning to real generative modeling. We construct a four-mode isotropic Gaussian in $\mathbb{R}^2$ and draw a reference pool of $N = 10^6$ points. The target centroid $\boldsymbol{T^*}$ is computed as the softmax-weighted centroid using the full pool, which at $N{=}10^6$ closely approximates the true expectation. For each sample size n, we subsample n points from the pool, compute $\boldsymbol{T_n}$, and repeat 200,000 times to estimate the bias norm $\|\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*}\|$ at each query, then average across 200 queries to summarize the pointwise scaling. Setup details are in Appendix F.

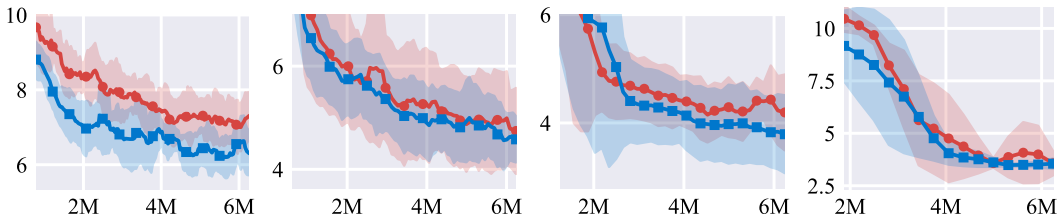


(a) $\tau = 0.1$ (b) $\tau = 0.2$

Figure 2: Bias norm $\|\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*}\|$ vs. n on a 2D four-mode Gaussian toy, log-log axes. Left: tight kernel (τ=0.1); right: wider kernel (τ=0.2). Standard traces slope -1 ($O(1/n)$), ABC slope -2 ($O(1/n^2)$).

In Figure 2, the Standard centroid traces a slope--1 line, and the ABC curve traces slope -2 over roughly two decades of n. At small n, e.g., n=4, 8, the two curves nearly overlap, because higher-order residuals in the ABC expansion are comparable in magnitude to the $O(1/n)$ bias of the Standard estimator when n is small. Even within this range, the ABC bias never exceeds the Standard bias at any n we tested. For larger n, the observed slopes closely match -1 and -2, empirically validating Theorems 3.3 and 4.1.

We also compare ABC with three classical bias-reduction baselines, namely the jackknife, a bootstrap correction, and the BR-SNIS estimator of Cardoso et al. [5], on a lighter CPU variant of this toy setup. Jackknife, Bootstrap, and ABC all operate on a fixed n-sample pool. Among them, ABC is the fastest correction while attaining comparable bias. BR-SNIS attains the smallest bias but requires roughly



(a) n=8 (b) n=16 (c) n=32 (d) n=64

Figure 3: FID vs. total positive samples seen (smoothed with a rolling average). Each panel fixes a per-class sample count n and compares the Standard variant of the drifting model (**red circles**) with its ABC-corrected counterpart (**blue squares**). Across all four values of n, the ABC curve lies below the baseline through most of training.

Kn fresh reference samples per estimate (with $K = 10$), a substantially larger sample budget than the others. Together, these results support ABC as the most efficient correction under the fixed-n sample budget typical of drifting-model training. The details are in Appendix G.

5.2 Image Generation Setup

We evaluate ABC on the DriftDiT-Small architecture [7], a 26.6M-parameter diffusion transformer that uses a frozen DINOv2 [21] multi-scale encoder and bidirectional kernel normalization. Main experiments use CIFAR-10 [16]. We sweep the per-class sample count $n \in \{8, 16, 32, 64\}$ used for both positive and negative centroids, where n=64 is the default setting of the original paper [7]. We train two variants that differ only in how the centroid is computed, the standard (uncorrected) estimator and the ABC-corrected one from Section 4.

Training. We follow the settings of Deng et al. [7], including the temperatures $\tau \in \{0.02, 0.05, 0.2\}$. The comparison is therefore mildly favorable to Standard, since the temperatures were chosen by Deng et al. [7] for the standard estimator and are not retuned for ABC. To equalize total positive samples seen across configurations, we scale training duration inversely with n so that n=64 trains for 500 epochs, n=32 for 1 000, etc. Full hyperparameters and computation details are in Appendix H.

Evaluation. We report Fréchet Inception Distance (FID) [12] computed on 50 k generated samples, using the same sampling configuration for Standard and ABC. All reported quantities are mean $\pm$ max-deviation across three random seeds unless noted otherwise.

5.3 Main Results

Figure 3 shows the FID trajectories throughout training for all four sample sizes, and Table 1 summarizes both the last-5-evaluation average and the best snapshot FID. ABC improves the last-5-average FID at every tested n, and the best FID in most cases. The largest gains occur at the smallest n, consistent with the $O(1/n)$ bias scaling of Theorem 3.3. This advantage is also visible in the smoothed trajectories of Figure 3. These observations match the picture from Section 3. They are consistent with the $O(1/n)$ pointwise bias having a measurable impact on generation quality, which ABC mitigates in the small-n regime. Single-seed results on CIFAR-100 presented in Appendix I also show that ABC improves both last-5 average and best FID, supporting consistency with the CIFAR-10 trends.

To complement FID, we additionally report the improved precision and recall of Kynkäänniemi et al. [17] in Appendix J. ABC improves recall rather than precision, consistent with the theoretical analysis of Theorem 3.3.

5.4 Training Speed

Beyond the endpoint FID, we ask how quickly each estimator reaches a target. Following the literature [24, 15], we record the first epoch at which FID drops below the threshold and convert to positive samples seen. As the 6.24 M-per-class budget is fixed across n, samples-to-FID is comparable across

Table 1: FID on CIFAR-10 for Standard vs ABC across per-class sample count n.

n	Total positives per class	Epochs	Last-5 average FID (↓)		Best FID (↓)	
			Standard	ABC	Standard	ABC
8	6.24 M	4 000	7.31 ± 0.94	$\mathbf{6.40 \pm 0.36}$ ↓0.91	6.83 ± 0.95	$\mathbf{5.81 \pm 0.34}$ ↓1.02
16	6.24 M	2 000	4.79 ± 1.04	$\mathbf{4.66 \pm 0.66}$ ↓0.13	4.46 ± 1.00	$\mathbf{4.30 \pm 0.50}$ ↓0.17
32	6.24 M	1 000	4.34 ± 0.60	$\mathbf{3.89 \pm 0.73}$ ↓0.45	4.02 ± 0.50	$\mathbf{3.71 \pm 0.69}$ ↓0.31
64	6.24 M	500	3.77 ± 0.89	$\mathbf{3.52 \pm 0.14}$ ↓0.24	$\mathbf{3.24 \pm 0.03}$	3.40 ± 0.08 ↑0.16

Table 2: Training samples (millions of positives, summed over all classes) required to first reach FID ≤ 10 and FID ≤ 7. $(k/3)$ flags attainment rate when fewer than three seeds reach it.

n	Required Samples to FID ≤ 10 (↓)		Required Samples to FID ≤ 7 (↓)	
	Standard	ABC	Standard	ABC
8	8.19 ± 4.29	$\mathbf{3.38 \pm 0.52}$ ↓4.81	26.52 (1/3)	$\mathbf{21.84 \pm 12.48}$ (3/3) ↓4.68
16	4.94 ± 1.04	$\mathbf{4.42 \pm 0.52}$ ↓0.52	11.44 ± 7.28	$\mathbf{7.80 \pm 4.68}$ ↓3.64
32	8.84 ± 3.64	8.84 ± 3.64	13.52 ± 5.20	$\mathbf{10.92 \pm 3.12}$ ↓2.60
64	16.64 ± 10.40	$\mathbf{13.52 \pm 8.32}$ ↓3.12	30.16 ± 4.16	$\mathbf{27.04 \pm 11.44}$ ↓3.12

batch sizes. Table 2 reports an attainment rate "$k/3$" and a sample-count summary over the seeds that reach the threshold. Despite the large variance inherent to first-hit-time measurements on noisy FID curves, ABC reaches the target with comparable or fewer samples at every n, with the strongest effect at small n. For example, at n=8 with FID ≤ 10, ABC needs about 40% of Standard's samples. With the tighter FID ≤ 7 threshold, only ABC reliably attains it across seeds ($3/3$ vs. $1/3$). The largest gains occur at the smallest n, consistent with the $O(1/n)$ bias shrinking with n.

5.5 Ablation Study on Positive vs. Negative Correction

ABC corrects both the positive centroid over real data and the negative centroid over generated samples. To isolate each contribution, we run an ablation at n=8 with four variants and report FID trajectories smoothed across seeds in Figure 4. First, every correction variant lies below the Standard curve, confirming that both sides of the centroid contribute bias that training can exploit. Second, the negative-only variant outperforms positive-only, consistent with self-masking reducing the negative side's effective sample count to $n-1$ and thereby increasing its $O(1/n)$ finite-sample bias relative to the positive side. Third, the fully combined ABC attains the lowest FID across most of training and is never clearly worse than any single-side variant. Taken together, these observations support adopting the combined correction as our default ABC deployment.

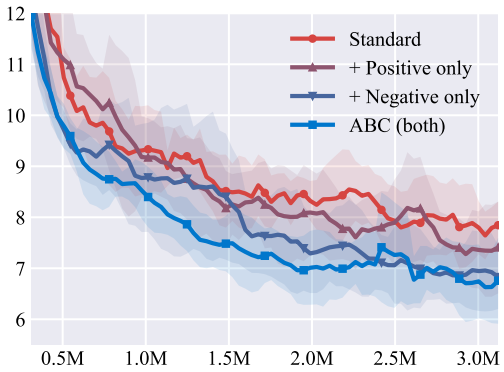


Figure 4: FID trajectories at n=8 with four variants. Bands show mean ± max-deviation across three seeds.

5.6 Computational Overhead

Table 3 compares the computational cost of ABC against the standard method at n=8. Epoch time is averaged over 10 epochs after startup. ABC introduces no additional parameters and negligible GPU memory overhead, since the correction operates entirely on the existing softmax weights via scalar arithmetic.

Table 3: Computational overhead comparison.

	Standard	ABC
Peak GPU memory (↓)	4 729 MB	4 731 MB ↑2MB
Epoch, no fusion (↓)	37.1 s	42.8 s ↑5.7s
Epoch, fused (↓)	38.0 s	37.0 s ↓1.0s
Parameters	26.6 M	26.6 M

Without kernel fusion, the per-epoch wall time increases from 37.1 s to 42.8 s due to the extra kernel launches for the squared-weight computation and the corrected centroid combination. However, with compiled execution via `torch.compile` the added operations are fused by the compiler into the surrounding kernel pipeline, and the wall-time difference from the standard method falls within the measurement noise of our 10-epoch average.

6 Conclusion

We identified a structural $O(1/n)$ subsampling bias in drifting models, arising from the softmax self-normalization of the minibatch estimator. We further proposed an Analytical Bias Correction (ABC) that reduces the leading-order bias from $O(1/n)$ to $O(1/n^2)$ while preserving convex-hull containment, and that can be implemented in two lines of code with negligible wall-time overhead under compiled execution. Experiments on CIFAR-10 confirmed that ABC delivers lower FID and faster training, with the largest gains at small n where the $O(1/n)$ bias is most significant.

Acknowledgments and Disclosure of Funding

This work used Anvil [25] at Purdue University through allocation CIS251316 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [3], which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. This research was also supported by the NVIDIA Academic Grant Program using its remote GPU server.

References

[1] E. M. L. Beale. Some Uses of Computers in Operational Research. *Industrielle Organisation*, 31(1):27–28, 1962.

[2] Yoshua Bengio and Jean-Sébastien Sénécal. Adaptive Importance Sampling to Accelerate Training of a Neural Probabilistic Language Model. *IEEE Transactions on Neural Networks*, 19(4):713–722, 2008.

[3] Timothy J Boerner, Stephen Deems, Thomas R Furlani, Shelley L Knuth, and John Towns. ACCESS: Advancing Innovation: NSF's Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support. In *PEARC*. 2023.

[4] Jiarui Cao, Zixuan Wei, and Yuxin Liu. Gradient Flow Drifting: Generative Modeling via Wasserstein Gradient Flows of KDE-Approximated Divergences. *arXiv preprint arXiv:2603.10592*, 2026.

[5] Gabriel Cardoso, Sergey Samsonov, Achille Thin, Eric Moulines, and Jimmy Olsson. BR-SNIS: Bias Reduced Self-Normalized Importance Sampling. In *NeurIPS*, 2022.

[6] William G. Cochran. *Sampling Techniques*. Wiley, 3rd edition, 1977.

[7] Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative Modeling via Drifting. *arXiv preprint arXiv:2602.04770*, 2026.

[8] Daniele Foffano, Alessio Russo, and Alexandre Proutiere. Receding-Horizon Control via Drifting Models. *arXiv preprint arXiv:2604.04528*, 2026.

[9] Leonard T. Franz, Sebastian Hoffmann, Tim Weiland, Bernhard Schölkopf, and Georg Martius. Drifting Fields are not Conservative. *arXiv preprint arXiv:2604.06333*, 2026.

[10] Ping He, Om Khangaonkar, Hamed Pirsiavash, Yikun Bai, and Soheil Kolouri. Sinkhorn-Drifting Generative Models. *arXiv preprint arXiv:2603.12366*, 2026.

[11] Tim Hesterberg. Weighted Average Importance Sampling and Defensive Mixture Distributions. *Technometrics*, 37(2):185–194, 1995.

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.

[13] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[14] Pipi Hu. Drifting to Boltzmann: Million-Fold Acceleration in Boltzmann Sampling with Force-Guided Drifting. *arXiv preprint arXiv:2603.05527*, 2026.

[15] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training Generative Adversarial Networks with Limited Data. In *NeurIPS*, 2020.

[16] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

[17] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved Precision and Recall Metric for Assessing Generative Models. In *NeurIPS*, 2019.

[18] Chieh-Hsin Lai, Bac Nguyen, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, Stefano Ermon, and Molei Tao. A Unified View of Score-Based and Drifting Models. *arXiv preprint arXiv:2603.07514*, 2026.

[19] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *ICLR*, 2023.

[20] Qing Lyu, Jianxu Wang, Jeremy Hudson, Ge Wang, and Chirstopher T. Whitlow. MRI-to-CT Synthesis Using Drifting Models. *arXiv preprint arXiv:2603.28498*, 2026.

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024.

[22] Art B. Owen. *Monte Carlo Theory, Methods and Examples*. Stanford, 2013. Chapter 9: Importance Sampling.

[23] M. H. Quenouille. Notes on Bias in Estimation. *Biometrika*, 43(3/4):353–360, 1956.

[24] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected GANs Converge Faster. In *NeurIPS*, 2021.

[25] X Carol Song, Preston Smith, Rajesh Kalyanam, Xiao Zhu, Eric Adams, Kevin Colby, Patrick Finnegan, Erik Gough, Elizabett Hillery, Rick Irvine, et al. Anvil — System Architecture and Experiences from Deployment and Early User Operations. In *PEARC*. 2022.

[26] M. Tin. Comparison of Some Ratio Estimators. *Journal of the American Statistical Association*, 60(309):294–307, 1965.

[27] Lin Zheng, Chong Wang, and Lingpeng Kong. Linear Complexity Randomized Self-Attention Mechanism. In *ICML*, 2022.

A Notation Summary

For ease of reference, we present a table of notations below.

Table 4: Table of notation.

Notation	Description
$\boldsymbol{x}$	Query point
$\boldsymbol{y}_i$	Reference point ($i = 1, \dots, n$)
n	Minibatch size
$w_i = k(\boldsymbol{x}, \boldsymbol{y}_i)$	Kernel weight
$\alpha_i = w_i / \sum_j w_j$	Normalized softmax weight
$\boldsymbol{T_n} = \sum_i \alpha_i \boldsymbol{y}_i$	Minibatch centroid (biased)
$\boldsymbol{T^*} = \mathbb{E}[w\boldsymbol{y}]/\mathbb{E}[w]$	Target centroid
$\Delta_n = \sum_i \alpha_i^2 (\boldsymbol{T_n} - \boldsymbol{y}_i)$	Plug-in bias correction
$\boldsymbol{T_n^{\mathrm{ABC}}} = \boldsymbol{T_n} - \Delta_n$	ABC-corrected centroid
$\mu_w = \mathbb{E}[w]$	Population mean kernel weight
$\boldsymbol{\mu}_v = \mathbb{E}[w\boldsymbol{y}]$	Population weighted mean
$\mathrm{Cov}(w, \boldsymbol{y}) = \mathbb{E}[(w - \mathbb{E}w)(\boldsymbol{y} - \mathbb{E}\boldsymbol{y})]$	Vector-valued covariance ($\in \mathbb{R}^D$)
$\mathrm{Var}(X) = \mathbb{E}\lVert X - \mathbb{E}X\rVert^2$	Variance (trace of covariance for vector X)
M, C	Boundedness constants ($w \le M$, $\lVert\boldsymbol{y}\rVert \le C$)
τ	Temperature parameter
D	Dimension

B Proofs for Section 3 (The Bias Problem)

B.1 Proof of Theorem 3.3 (Leading-Order Bias)

Theorem 3.3 (Leading-order bias, restated). *Let $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n \overset{\text{i.i.d.}}{\sim} p$ in $\mathbb{R}^D$ with $w = k(\boldsymbol{x}, \boldsymbol{y}) \in [0, M]$ and $\lVert\boldsymbol{y}\rVert \le C$ almost surely for some constants $M, C > 0$, and $\Pr(w > 0) = 1$. Then, for fixed $\boldsymbol{x}$ with $\mu_w := \mathbb{E}[w] > 0$,*

$$\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*} \;=\; -\frac{1}{n}\,\frac{\mathbb{E}\big[w^2(\boldsymbol{y} - \boldsymbol{T^*})\big]}{\mu_w^2} \;+\; O(n^{-2}).$$

Proof. We give the full proof using the multivariate delta method, which makes the remainder control explicit.

Setup. Recall $\boldsymbol{T_n} = \bar{\boldsymbol{v}}/\bar{w}$, where $\bar{\boldsymbol{v}} = n^{-1}\sum_{i=1}^n w_i \boldsymbol{y}_i$ and $\bar{w} = n^{-1}\sum_{i=1}^n w_i$. Define the full-distribution expectations $\mu_w := \mathbb{E}[w]$, $\boldsymbol{\mu}_v := \mathbb{E}[w\boldsymbol{y}]$, and the target $\boldsymbol{T^*} := \boldsymbol{\mu}_v/\mu_w$. For the vector-valued map $g(\boldsymbol{v}, w) = \boldsymbol{v}/w$, partial derivatives in $\boldsymbol{v}$ are linear maps on $\mathbb{R}^D$ (e.g., $\partial g/\partial \boldsymbol{v}$ acts as $(1/\mu_w)\, I_D$), and $\mathrm{Cov}(w\boldsymbol{y}, w)$ denotes the componentwise vector $\mathbb{E}[(w\boldsymbol{y} - \boldsymbol{\mu}_v)(w - \mu_w)] \in \mathbb{R}^D$.

Step 1. Define the smooth map and expand. Let $g(\boldsymbol{v}, w) := \boldsymbol{v}/w$, so that $\boldsymbol{T_n} = g(\bar{\boldsymbol{v}}, \bar{w})$ and $\boldsymbol{T^*} = g(\boldsymbol{\mu}_v, \mu_w)$. Taylor-expanding g around $(\boldsymbol{\mu}_v, \mu_w)$ to second order,

$$\begin{aligned}\boldsymbol{T_n} = g(\boldsymbol{\mu}_v, \mu_w) &+ \frac{\partial g}{\partial \boldsymbol{v}}\,(\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v) + \frac{\partial g}{\partial w}\,(\bar{w} - \mu_w) \\ &+ \frac{1}{2}\,\frac{\partial^2 g}{\partial w^2}\,(\bar{w} - \mu_w)^2 + \frac{\partial^2 g}{\partial \boldsymbol{v}\,\partial w}\,(\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v)(\bar{w} - \mu_w) + R_n, \end{aligned} \tag{11}$$

where $R_n := \boldsymbol{T_n} - P_2(\bar{\boldsymbol{v}}, \bar{w})$ is the exact second-order Taylor residual, with P_2 the degree-2 Taylor polynomial of g at $(\boldsymbol{\mu}_v, \mu_w)$. We show in the "Remainder bound" paragraph below that $\lVert\mathbb{E}[R_n]\rVert = O(n^{-2})$ under the stated assumptions.

Step 2. Compute partial derivatives. For $g(\boldsymbol{v}, w) = \boldsymbol{v}/w$, evaluating at $(\boldsymbol{\mu}_v, \mu_w)$ yields

$$\frac{\partial g}{\partial \boldsymbol{v}} = \frac{I_D}{\mu_w}, \quad \frac{\partial g}{\partial w} = -\frac{\boldsymbol{\mu}_v}{\mu_w^2} = -\frac{\boldsymbol{T}^*}{\mu_w}, \quad \frac{\partial^2 g}{\partial w^2} = \frac{2\boldsymbol{\mu}_v}{\mu_w^3} = \frac{2\boldsymbol{T}^*}{\mu_w^2}, \quad \frac{\partial^2 g}{\partial \boldsymbol{v}\,\partial w} = -\frac{I_D}{\mu_w^2}. \tag{12}$$

Note that $\partial^2 g/\partial \boldsymbol{v}^2 = 0$ (linearity in $\boldsymbol{v}$), so no quadratic-in-$\boldsymbol{v}$ term appears in (11).

Step 3. Take expectations. The first-order terms vanish since $\mathbb{E}[\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v] = \mathbf{0}$ and $\mathbb{E}[\bar{w} - \mu_w] = 0$. The bias thus comes entirely from the second-order terms,

$$\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T}^* = \frac{\boldsymbol{T}^*}{\mu_w^2}\,\mathbb{E}\big[(\bar{w} - \mu_w)^2\big] - \frac{1}{\mu_w^2}\,\mathbb{E}\big[(\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v)(\bar{w} - \mu_w)\big] + O(n^{-2}). \tag{13}$$

Step 4. Evaluate covariances using the i.i.d. structure. Since $w_1, \ldots, w_n$ are i.i.d., we have

$$\mathbb{E}\big[(\bar{w} - \mu_w)^2\big] = \mathrm{Var}(\bar{w}) = \frac{1}{n}\,\mathrm{Var}(w) = \frac{1}{n}\big(\mathbb{E}[w^2] - \mu_w^2\big), \tag{14}$$

$$\mathbb{E}\big[(\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v)(\bar{w} - \mu_w)\big] = \frac{1}{n}\,\mathrm{Cov}(w\boldsymbol{y},\, w) = \frac{1}{n}\big(\mathbb{E}[w^2\boldsymbol{y}] - \boldsymbol{\mu}_v\,\mu_w\big). \tag{15}$$

The cross-covariance reduces to a single-sample quantity because for $i \neq j$ independence gives $\mathbb{E}[(w_i\boldsymbol{y}_i - \boldsymbol{\mu}_v)(w_j - \mu_w)] = \mathbf{0}$.

Step 5. Simplify to the displacement form. Substituting (14) and (15) into (13) yields

$$\begin{aligned}
\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T}^* &= \frac{1}{n\mu_w^2}\Big[\boldsymbol{T}^*\big(\mathbb{E}[w^2] - \mu_w^2\big) - \big(\mathbb{E}[w^2\boldsymbol{y}] - \boldsymbol{\mu}_v\,\mu_w\big)\Big] + O(n^{-2}) \\
&= \frac{1}{n\mu_w^2}\Big[\boldsymbol{T}^*\,\mathbb{E}[w^2] - \mathbb{E}[w^2\boldsymbol{y}] \underbrace{-\,\boldsymbol{T}^*\mu_w^2 + \boldsymbol{\mu}_v\,\mu_w}_{=0 \text{ since } \boldsymbol{\mu}_v = \mu_w \boldsymbol{T}^*}\Big] + O(n^{-2}) \\
&= -\frac{1}{n\mu_w^2}\,\mathbb{E}\big[w^2(\boldsymbol{y} - \boldsymbol{T}^*)\big] + O(n^{-2}).
\end{aligned} \tag{16}$$

The cancellation in the second line is exact, since $\boldsymbol{\mu}_v\,\mu_w - \boldsymbol{T}^*\mu_w^2 = \mu_w(\boldsymbol{\mu}_v - \boldsymbol{T}^*\mu_w) = \mathbf{0}$ by the definition $\boldsymbol{T}^* = \boldsymbol{\mu}_v/\mu_w$.

Remainder bound. We show $\|\mathbb{E}[R_n]\| = O(n^{-2})$ by splitting $\mathbb{E}[R_n] = \mathbb{E}[R_n\,\mathbf{1}_{G_n}] + \mathbb{E}[R_n\,\mathbf{1}_{G_n^c}]$ on the good event $G_n := \{\bar{w} \geq \mu_w/2\}$. On G_n, the map $g(\boldsymbol{v}, w) = \boldsymbol{v}/w$ is smooth with uniformly bounded partials, while G_n^c has super-polynomially small probability.

Step 1. Dispose of G_n^c. Since $w \in [0, M]$, Hoeffding's inequality [13] with $t = \mu_w/2$ gives

$$\Pr(G_n^c) \;\leq\; 2\,\exp\!\left(-\frac{n\,\mu_w^2}{2M^2}\right), \tag{17}$$

which decays faster than any polynomial in n.

The assumption $\Pr(w > 0) = 1$ gives $\bar{w} > 0$ a.s., so $\boldsymbol{T_n}$ is a convex combination of the $\boldsymbol{y}_i$ with $\|\boldsymbol{T_n}\| \leq C$. Meanwhile P_2 is a fixed polynomial in $(\bar{\boldsymbol{v}}, \bar{w})$ with coefficients that are rational functions of $(\boldsymbol{\mu}_v, \mu_w)$, finite since $\mu_w > 0$, evaluated on the bounded region $\|\bar{\boldsymbol{v}}\| \leq MC$, $\bar{w} \leq M$. Therefore both $\boldsymbol{T_n}$ and $P_2(\bar{\boldsymbol{v}}, \bar{w})$ are a.s. bounded. Triangle inequality then gives $\|R_n\| \leq \|\boldsymbol{T_n}\| + \|P_2\| \leq C_0$ a.s. for some $C_0(M, C, \mu_w, \|\boldsymbol{T}^*\|)$, and therefore

$$\|\mathbb{E}[R_n\,\mathbf{1}_{G_n^c}]\| \;\leq\; C_0\,\Pr(G_n^c) \;=\; O(n^{-2}).$$

Step 2. Taylor expansion on G_n. Let $\boldsymbol{Z}_i := (w_i\boldsymbol{y}_i - \boldsymbol{\mu}_v,\, w_i - \mu_w) \in \mathbb{R}^{D+1}$ denote the centred i-th sample contribution ($\mathbb{E}[\boldsymbol{Z}_i] = \mathbf{0}$), and write $\boldsymbol{\xi}_n := (\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v,\, \bar{w} - \mu_w) = n^{-1}\sum_{i=1}^n \boldsymbol{Z}_i$, with coordinates indexed by $\ell \in \{1, \ldots, D+1\}$. By Taylor's theorem, expanded to fourth order,

$$R_n \;=\; \underbrace{\tfrac{1}{6}\sum_{\ell_1,\ell_2,\ell_3} \partial^3_{\ell_1\ell_2\ell_3} g(\boldsymbol{\mu}_v, \mu_w)\;\xi_n^{(\ell_1)}\xi_n^{(\ell_2)}\xi_n^{(\ell_3)}}_{=:\,R_n^{(3)}} \;+\; R_n^{(4)}, \tag{18}$$

where the third-order partials are evaluated at $(\boldsymbol{\mu}_v, \mu_w)$, and $R_n^{(4)}$ is the fourth-order remainder.

Every k-th partial of $g(\boldsymbol{v}, w) = \boldsymbol{v}/w$ is a polynomial in $\boldsymbol{v}$ divided by w^{k+1}, so on G_n (where $\bar{w} \geq \mu_w/2$ and $\|\boldsymbol{v}\| \leq MC$) all third- and fourth-order partials along the segment are uniformly bounded.

Step 3. Cubic and quartic expectations are $O(n^{-2})$. Recall from Step 2 that $\xi_n^{(\ell)} = n^{-1}\sum_{i=1}^n Z_i^{(\ell)}$ with $\mathbb{E}[Z_i^{(\ell)}] = 0$ and bounded fourth moments (from $w \in [0, M]$ and $\|\boldsymbol{y}\| \leq C$). A direct index count then gives the following.

- **Cubic.** $\mathbb{E}[\xi_n^{(\ell_1)}\xi_n^{(\ell_2)}\xi_n^{(\ell_3)}] = \frac{1}{n^3}\sum_{i,j,k}\mathbb{E}[Z_i^{(\ell_1)}Z_j^{(\ell_2)}Z_k^{(\ell_3)}] = \frac{1}{n^3}\cdot n\cdot\mathbb{E}[Z_1^{(\ell_1)}Z_1^{(\ell_2)}Z_1^{(\ell_3)}] = O(n^{-2})$, since terms with any non-matching index factor through $\mathbb{E}[Z_i^{(\ell)}] = 0$. Only the n diagonal terms $i = j = k$ survive, each a bounded third cross-moment.

  Since $R_n^{(3)}$ has sample-independent coefficients, summing the cubic moments over the $(D+1)^3$ index triples gives $\|\mathbb{E}[R_n^{(3)}]\| = O(n^{-2})$. Writing $\mathbb{E}[R_n^{(3)}\,\mathbf{1}_{G_n}] = \mathbb{E}[R_n^{(3)}] - \mathbb{E}[R_n^{(3)}\,\mathbf{1}_{G_n^c}]$ and noting that $R_n^{(3)}$ is a.s. bounded (a polynomial in the a.s. bounded $\boldsymbol{\xi}_n$), the second term is $O(\Pr(G_n^c)) = O(n^{-2})$. Hence $\|\mathbb{E}[R_n^{(3)}\,\mathbf{1}_{G_n}]\| = O(n^{-2})$.

- **Quartic.** By a coordinate-wise Lagrange remainder form, each component of $R_n^{(4)}$ equals $\frac{1}{4!}\sum_{\ell_1,\ell_2,\ell_3,\ell_4}\partial^4_{\ell_1\ell_2\ell_3\ell_4}g(c_n)\,\xi_n^{(\ell_1)}\xi_n^{(\ell_2)}\xi_n^{(\ell_3)}\xi_n^{(\ell_4)}$ for some c_n (allowed to differ per coordinate) on the segment joining $(\boldsymbol{\mu}_v, \mu_w)$ to $(\bar{\boldsymbol{v}}, \bar{w})$. On G_n, the w-coordinate of c_n is $\geq \mu_w/2$, so the fourth-order partials along the segment are uniformly bounded, giving $\|R_n^{(4)}\mathbf{1}_{G_n}\| = O(\|\boldsymbol{\xi}_n\|^4)$ pointwise. Each fourth-order monomial $\mathbb{E}[\xi_n^{(\ell_1)}\cdots\xi_n^{(\ell_4)}]$ is $O(n^{-2})$, since terms with any singleton index vanish by centring and the survivors (two matched pairs, $O(n^2)$ choices each $O(n^{-4})$, or all four matched, n choices each $O(n^{-4})$) each contribute $O(n^{-2})$. By Cauchy–Schwarz, $\|\boldsymbol{\xi}_n\|^4 \leq (D+1)\sum_\ell(\xi_n^{(\ell)})^4$, so $\mathbb{E}\|\boldsymbol{\xi}_n\|^4 = O(n^{-2})$. Jensen's inequality $\|\mathbb{E}[X]\| \leq \mathbb{E}\|X\|$ then gives $\|\mathbb{E}[R_n^{(4)}\mathbf{1}_{G_n}]\| \leq \mathbb{E}\|R_n^{(4)}\mathbf{1}_{G_n}\| = O(\mathbb{E}\|\boldsymbol{\xi}_n\|^4) = O(n^{-2})$.

Combining the three bounds above, $\|\mathbb{E}[R_n]\| \leq C_R/n^2$ for a constant C_R depending on M, C, μ_w, D, and $\|\boldsymbol{T}^*\|$. □

C Proofs for Section 4 (Analytical Bias Correction)

C.1 Proof of Theorem 4.1 (Bias Reduction)

Theorem 4.1 (Bias reduction, restated). *Under the assumptions of Theorem 3.3 (i.e., $w \in [0, M]$ a.s., $\|\boldsymbol{y}\| \leq C$ a.s., $\Pr(w > 0) = 1$, $\mathbb{E}[w] > 0$):*

$$\mathbb{E}\big[\boldsymbol{T}_n^{\mathbf{ABC}} - \boldsymbol{T}^*\big] = O(n^{-2}).$$

Proof. Since $\boldsymbol{T}_n^{\mathbf{ABC}} = \boldsymbol{T}_n - \Delta_n$, it suffices to show

$$\mathbb{E}[\Delta_n] \;=\; \big(\mathbb{E}[\boldsymbol{T}_n] - \boldsymbol{T}^*\big) + O(n^{-2}). \tag{19}$$

Then $\mathbb{E}[\boldsymbol{T}_n^{\mathbf{ABC}} - \boldsymbol{T}^*] = (\mathbb{E}[\boldsymbol{T}_n] - \boldsymbol{T}^*) - \mathbb{E}[\Delta_n] = O(n^{-2})$.

All Taylor expansions and delta-method bounds below are made rigorous by the good-event splitting $G_n = \{\bar{w} \geq \mu_w/2\}$ introduced in the proof of Theorem 3.3. On G_n the maps $1/\bar{w}^k$ and $g(u, w) = u/w^2$ are Lipschitz with constants depending only on M and μ_w, so Taylor remainders and delta-method variance bounds hold deterministically. Off G_n the relevant quantities are uniformly bounded, and $\Pr(G_n^c)$ is super-polynomially small by Hoeffding, so off-G_n contributions are absorbed into the $O(n^{-2})$ remainder. We write the expansions pointwise for brevity.

Step 1. Decompose Δ_n. Writing $\boldsymbol{T_n} - \boldsymbol{y}_i = (\boldsymbol{T_n} - \boldsymbol{T^*}) + (\boldsymbol{T^*} - \boldsymbol{y}_i)$ and summing with weights α_i^2:

$$\Delta_n \;=\; \underbrace{\sum_{i=1}^{n} \alpha_i^2 (\boldsymbol{T^*} - \boldsymbol{y}_i)}_{(A)} \;+\; \underbrace{\Big(\sum_{i=1}^{n} \alpha_i^2\Big)(\boldsymbol{T_n} - \boldsymbol{T^*})}_{(B)}. \tag{20}$$

We show $\mathbb{E}[(A)] = (\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*}) + O(n^{-2})$ and $\mathbb{E}[(B)] = O(n^{-2})$.

Step 2. Analyze term (A). Writing $\alpha_i^2 = w_i^2/(\sum_j w_j)^2 = w_i^2/(n\bar{w})^2$ with $\bar{w} = n^{-1}\sum_j w_j$:

$$(A) \;=\; \frac{1}{\bar{w}^2} \cdot \underbrace{\frac{1}{n} \cdot \frac{1}{n} \sum_{i=1}^{n} w_i^2 (\boldsymbol{T^*} - \boldsymbol{y}_i)}_{= (1/n)\,\bar{U}_n}, \qquad \bar{U}_n := \frac{1}{n}\sum_{i=1}^{n} w_i^2 (\boldsymbol{T^*} - \boldsymbol{y}_i).$$

Taylor-expand $1/\bar{w}^2$ around μ_w with $\epsilon := \bar{w} - \mu_w = O_p(n^{-1/2})$ by Chebyshev's inequality:

$$\frac{1}{\bar{w}^2} \;=\; \frac{1}{\mu_w^2}\Big(1 - \frac{2\epsilon}{\mu_w} + O_p(\epsilon^2)\Big).$$

Multiplying:

$$(A) \;=\; \frac{\bar{U}_n}{n\mu_w^2} \;-\; \frac{2\,\bar{U}_n\,\epsilon}{n\mu_w^3} \;+\; O_p\Big(\frac{\|\bar{U}_n\|\,\epsilon^2}{n}\Big).$$

We take expectations term by term.

- *Leading term.* $\mathbb{E}[\bar{U}_n] = \mathbb{E}[w^2(\boldsymbol{T^*} - \boldsymbol{y})] = -\mathbb{E}[w^2(\boldsymbol{y} - \boldsymbol{T^*})]$. So the first term equals $-\mathbb{E}[w^2(\boldsymbol{y} - \boldsymbol{T^*})]/(n\mu_w^2)$. By Theorem 3.3, this equals $\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*} + O(n^{-2})$.
- *Cross term.* Expanding $\bar{U}_n\,\epsilon = n^{-2}\sum_{i,j} w_i^2(\boldsymbol{T^*} - \boldsymbol{y}_i)(w_j - \mu_w)$, the $i \neq j$ terms vanish by independence (second factor is centred), leaving $\mathbb{E}[\bar{U}_n\,\epsilon] = n^{-1}\,\mathbb{E}[w^2(\boldsymbol{T^*} - \boldsymbol{y})(w - \mu_w)] = n^{-1}\,\mathrm{Cov}\big(w^2(\boldsymbol{T^*} - \boldsymbol{y}),\, w\big) = O(n^{-1})$. Contribution $O(n^{-1})/n = O(n^{-2})$.
- *Higher-order term.* Under $w \in [0, M]$ and $\|\boldsymbol{y}\| \le C$, $\|\bar{U}_n\| \le M^2(\|\boldsymbol{T^*}\| + C)$ a.s. (as $\bar{U}_n = n^{-1}\sum_i w_i^2(\boldsymbol{T^*} - \boldsymbol{y}_i)$ is bounded uniformly), so $\mathbb{E}[\|\bar{U}_n\|\,\epsilon^2] \le M^2(\|\boldsymbol{T^*}\| + C)\,\mathbb{E}[\epsilon^2] = O(n^{-1})$ since $\mathbb{E}[\epsilon^2] = \mathrm{Var}(\bar{w}) = O(n^{-1})$. Contribution $O(n^{-2})$.

Combining, $\mathbb{E}[(A)] = (\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*}) + O(n^{-2})$.

Step 3. Analyze term (B). Let $A_n := \sum_i \alpha_i^2$ and $B_n := \boldsymbol{T_n} - \boldsymbol{T^*}$. Decompose the expectation of a product:

$$\mathbb{E}[(B)] \;=\; \mathbb{E}[A_n]\,\mathbb{E}[B_n] \;+\; \mathrm{Cov}(A_n, B_n).$$

Mean-product. Since $w_i \le M$, $w_i^2 \le M w_i$, hence $\sum_i w_i^2 \le M\sum_i w_i$. Therefore $A_n = \frac{\sum_i w_i^2}{(\sum_i w_i)^2} \le \frac{M}{\sum_i w_i}$. On the good event $G_n = \{\bar{w} \ge \mu_w/2\}$ this gives $A_n \mathbf{1}_{G_n} \le 2M/(n\mu_w)$. Off G_n, $A_n \le 1$ trivially, and with $\Pr(G_n^c) = O(n^{-k})$ for all k from (17), $\mathbb{E}[A_n] = O(n^{-1})$. By Theorem 3.3, $\mathbb{E}[B_n] = O(n^{-1})$. Hence $\mathbb{E}[A_n]\,\mathbb{E}[B_n] = O(n^{-2})$.

Covariance. Since B_n is vector-valued, apply Cauchy–Schwarz componentwise: $\|\mathrm{Cov}(A_n, B_n)\| \le \sqrt{\mathrm{Var}(A_n)}\,\sqrt{\mathbb{E}\|B_n - \mathbb{E}B_n\|^2}$. Writing $A_n = (1/n)\,g(\bar{u}_n, \bar{w})$ with $\bar{u}_n = n^{-1}\sum_i w_i^2$ and $g(u, w) = u/w^2$ Lipschitz on G_n, the delta method gives $\mathrm{Var}(A_n) = O(n^{-3})$. Likewise $\boldsymbol{T_n} = \bar{\boldsymbol{v}}/\bar{w}$ is Lipschitz on G_n, so $\mathbb{E}\|B_n - \mathbb{E}B_n\|^2 = O(n^{-1})$. Therefore $\|\mathrm{Cov}(A_n, B_n)\| = O(n^{-2})$.

Combining, $\mathbb{E}[(B)] = O(n^{-2})$.

Step 4. Conclude. From Steps 2 and 3, $\mathbb{E}[\Delta_n] = \mathbb{E}[(A)] + \mathbb{E}[(B)] = (\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*}) + O(n^{-2})$, which is (19). Therefore $\mathbb{E}[\boldsymbol{T_n^{\mathrm{ABC}}} - \boldsymbol{T^*}] = O(n^{-2})$. □

C.2 Proof of Proposition 4.3 (No First-Order Total-Variance Increase)

Proposition 4.3 (No first-order total-variance increase, restated). *Under the conditions of Theorem 4.1 (i.e., $w \in [0, M]$ a.s., $\|\boldsymbol{y}\| \le C$ a.s., $\Pr(w > 0) = 1$, $\mathbb{E}[w] > 0$):*

$$\mathbb{E}\big[\|\boldsymbol{T}_n^{\mathbf{ABC}} - \mathbb{E}[\boldsymbol{T}_n^{\mathbf{ABC}}]\|^2\big] \;=\; \mathbb{E}\big[\|\boldsymbol{T}_n - \mathbb{E}[\boldsymbol{T}_n]\|^2\big] + O(n^{-3/2}) \;=\; O(n^{-1}).$$

Proof. In this proof, $\mathrm{Var}(X)$ denotes $\mathbb{E}\|X - \mathbb{E}[X]\|^2$. This equals the trace of the covariance matrix for a vector X (equivalently $\sum_d \mathrm{Var}(X_d)$) and the usual variance for a scalar X. We first bound $\mathrm{Var}(\boldsymbol{T}_n)$, then decompose $\mathrm{Var}(\boldsymbol{T}_n^{\mathbf{ABC}})$ via $\boldsymbol{T}_n^{\mathbf{ABC}} = \boldsymbol{T}_n - \Delta_n$.

Step 0. Baseline variance. By the optimality of the mean in L^2, $\mathrm{Var}(\boldsymbol{T}_n) = \mathbb{E}[\|\boldsymbol{T}_n - \mathbb{E}[\boldsymbol{T}_n]\|^2] \le \mathbb{E}[\|\boldsymbol{T}_n - \boldsymbol{T}^*\|^2]$. It therefore suffices to show $\mathbb{E}[\|\boldsymbol{T}_n - \boldsymbol{T}^*\|^2] = O(n^{-1})$.

On G_n. On $G_n = \{\bar{w} \ge \mu_w/2\}$, since $\|\bar{\boldsymbol{v}}\| \le C\bar{w} \le CM$ and $\|\boldsymbol{\mu}_v\| \le CM$, the Taylor segment from $(\boldsymbol{\mu}_v, \mu_w)$ to $(\bar{\boldsymbol{v}}, \bar{w})$ lies in a region where $g(\boldsymbol{v}, w) = \boldsymbol{v}/w$ has uniformly bounded second derivatives. The first-order Taylor expansion around $(\boldsymbol{\mu}_v, \mu_w)$ gives

$$\boldsymbol{T}_n - \boldsymbol{T}^* = \frac{\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v}{\mu_w} - \frac{\boldsymbol{T}^*(\bar{w} - \mu_w)}{\mu_w} + R_n',$$

with $\|R_n'\| \le C_1\big(\|\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v\| + |\bar{w} - \mu_w|\big)^2$ uniformly on G_n for a constant C_1 independent of n. Centred i.i.d. sample means of bounded random variables have fourth central moments $O(n^{-2})$, so $\mathbb{E}[(\|\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v\| + |\bar{w} - \mu_w|)^4] = O(n^{-2})$ and hence $\mathbb{E}[\|R_n'\|^2\,\mathbf{1}_{G_n}] = O(n^{-2})$. Combined with $\mathbb{E}\|\bar{\boldsymbol{v}} - \boldsymbol{\mu}_v\|^2, \mathbb{E}(\bar{w} - \mu_w)^2 = O(n^{-1})$, the bound $\|a + b\|^2 \le 2(\|a\|^2 + \|b\|^2)$ yields

$$\mathbb{E}\big[\|\boldsymbol{T}_n - \boldsymbol{T}^*\|^2\,\mathbf{1}_{G_n}\big] = O(n^{-1}).$$

Off G_n. Since $\|\boldsymbol{y}\| \le C$ a.s. and $\boldsymbol{T}_n$ is a convex combination of the $\boldsymbol{y}_i$, $\|\boldsymbol{T}_n\| \le C$ and hence $\|\boldsymbol{T}_n - \boldsymbol{T}^*\| \le C + \|\boldsymbol{T}^*\|$. By (17), $\mathbb{E}[\|\boldsymbol{T}_n - \boldsymbol{T}^*\|^2\,\mathbf{1}_{G_n^c}] \le (C + \|\boldsymbol{T}^*\|)^2\,\Pr(G_n^c) = o(n^{-k})$ for every $k \ge 1$.

Combining, $\mathrm{Var}(\boldsymbol{T}_n) \le \mathbb{E}[\|\boldsymbol{T}_n - \boldsymbol{T}^*\|^2] = O(n^{-1})$.

Since $\boldsymbol{T}_n^{\mathbf{ABC}} = \boldsymbol{T}_n - \Delta_n$,

$$\mathrm{Var}(\boldsymbol{T}_n^{\mathbf{ABC}}) = \mathrm{Var}(\boldsymbol{T}_n) - 2\,\mathbb{E}\big[\langle \boldsymbol{T}_n - \mathbb{E}[\boldsymbol{T}_n],\; \Delta_n - \mathbb{E}[\Delta_n]\rangle\big] + \mathrm{Var}(\Delta_n). \tag{21}$$

It suffices to show that both $\mathrm{Var}(\Delta_n)$ and the cross term are $O(n^{-3/2})$.

Step 1. Correction variance. Write $A_n := \sum_i \alpha_i^2 = (\sum_i w_i^2)/(\sum_i w_i)^2$, as also defined in the proof of Theorem 4.1. We first show

$$\mathbb{E}[A_n^2] \;=\; O(n^{-2}). \tag{22}$$

From the proof of Theorem 4.1, $A_n \mathbf{1}_{G_n} \le 2M/(n\mu_w)$, so $\mathbb{E}[A_n^2\,\mathbf{1}_{G_n}] \le 4M^2/(n^2\mu_w^2) = O(n^{-2})$. Off G_n, $A_n \le \sum_i \alpha_i = 1$, so $\mathbb{E}[A_n^2\,\mathbf{1}_{G_n^c}] \le \Pr(G_n^c) = o(n^{-k})$ for any k by (17). Combining yields (22).

Since $\boldsymbol{T}_n$ lies in the convex hull of $\{\boldsymbol{y}_i\}$, each $\boldsymbol{y}_i - \boldsymbol{T}_n$ is bounded by $\|\boldsymbol{y}_i\| + \|\boldsymbol{T}_n\| \le 2C$ almost surely under the assumption $\|\boldsymbol{y}\| \le C$. Applying the triangle inequality to $\Delta_n = \sum_i \alpha_i^2(\boldsymbol{T}_n - \boldsymbol{y}_i)$,

$$\|\Delta_n\| \;\le\; \sum_i \alpha_i^2\,\|\boldsymbol{y}_i - \boldsymbol{T}_n\| \;\le\; 2C\,A_n,$$

so $\|\Delta_n\|^2 \le 4C^2\,A_n^2$. Taking expectations and using (22),

$$\mathbb{E}\big[\|\Delta_n\|^2\big] \;\le\; 4C^2\,\mathbb{E}[A_n^2] \;=\; O(n^{-2}). \tag{23}$$

Hence $\mathrm{Var}(\Delta_n) \le \mathbb{E}[\|\Delta_n\|^2] = O(n^{-2})$.

Step 2. Bound the cross term. By the Cauchy–Schwarz inequality,

$$\begin{aligned}\left|\mathbb{E}\langle \boldsymbol{T_n} - \mathbb{E}[\boldsymbol{T_n}],\, \Delta_n - \mathbb{E}[\Delta_n]\rangle\right| &\le \sqrt{\mathrm{Var}(\boldsymbol{T_n}) \cdot \mathrm{Var}(\Delta_n)} \\ &\le \sqrt{O(n^{-1}) \cdot O(n^{-2})} \;=\; O(n^{-3/2}).\end{aligned}$$

Step 3. Combine. Substituting into (21),

$$\mathrm{Var}(\boldsymbol{T}_n^{\mathbf{ABC}}) = \mathrm{Var}(\boldsymbol{T_n}) + O(n^{-3/2}) + O(n^{-2}) = \mathrm{Var}(\boldsymbol{T_n}) + O(n^{-3/2}).$$

Since $\mathrm{Var}(\boldsymbol{T_n}) = O(n^{-1})$, the $O(n^{-3/2})$ remainder is of smaller order, confirming that ABC preserves the same $O(n^{-1})$ leading-order variance. □

Intuition. The correction Δ_n has magnitude $O(1/n)$, the same order as the bias it removes. The statistical fluctuations of $\boldsymbol{T_n}$ are $O(1/\sqrt{n})$, which is a factor of $\sqrt{n}$ larger. Subtracting an $O(1/n)$ term from an $O(1/\sqrt{n})$-noisy estimator shifts the mean (reducing bias) without meaningfully changing the spread (variance).

D Temperature Cannot Substitute for Bias Correction

Since both the subsampling bias and high temperature push the centroid toward the global mean, a natural question is whether simply lowering τ can compensate for the bias, removing the need for an analytical correction. Let $\boldsymbol{T_n}(\boldsymbol{x}; \tau)$ denote the minibatch centroid estimator (4) computed at temperature τ from n i.i.d. samples. The question asks whether there exists a $\tau' \neq \tau$ such that

$$\mathbb{E}[\boldsymbol{T_n}(\boldsymbol{x}; \tau')] \;=\; \boldsymbol{T}^*(\boldsymbol{x}; \tau) \qquad \forall\, \boldsymbol{x},$$

i.e., whether training at a modified temperature can exactly undo the bias one would incur at the intended temperature.

D.1 The Compensation Equation

Write $k_\tau(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|_2/\tau)$, $w = k_\tau(\boldsymbol{x}, \boldsymbol{y})$, and $\boldsymbol{T}^*(\boldsymbol{x}; \tau) = \mathbb{E}_\tau[w\boldsymbol{y}]/\mathbb{E}_\tau[w]$ to make the τ-dependence explicit. By Theorem 3.3, the biased estimator satisfies

$$\mathbb{E}[\boldsymbol{T_n}(\boldsymbol{x}; \tau)] \;=\; \boldsymbol{T}^*(\boldsymbol{x}; \tau) \;-\; \frac{1}{n}\,\frac{\mathbb{E}_\tau[w^2(\boldsymbol{y} - \boldsymbol{T}^*)]}{(\mathbb{E}_\tau[w])^2} \;+\; O(n^{-2}).$$

For a compensating shift $\tau' = \tau + \Delta$ with $\Delta = O(n^{-1})$, first-order matching $\mathbb{E}[\boldsymbol{T_n}(\boldsymbol{x}; \tau')] = \boldsymbol{T}^*(\boldsymbol{x}; \tau)$ requires

$$\partial_\tau \boldsymbol{T}^*(\boldsymbol{x}; \tau)\; \Delta \;=\; \frac{1}{n}\,\frac{\mathbb{E}_\tau[w^2(\boldsymbol{y} - \boldsymbol{T}^*)]}{(\mathbb{E}_\tau[w])^2} \tag{24}$$

simultaneously for every $\boldsymbol{x}$. Since Δ is a single scalar independent of $\boldsymbol{x}$, (24) imposes two conditions in general dimension D:

1. $\partial_\tau \boldsymbol{T}^*(\boldsymbol{x}; \tau)$ and $\mathbb{E}_\tau[w^2(\boldsymbol{y} - \boldsymbol{T}^*)]$ are *collinear* at every $\boldsymbol{x}$ (vacuous when $D = 1$);
2. their proportionality constant is $\boldsymbol{x}$-independent. Equivalently, when $D = 1$, the scalar ratio

$$\frac{\mathbb{E}_\tau[w^2(\boldsymbol{y} - \boldsymbol{T}^*)]}{(\mathbb{E}_\tau[w])^2\, \partial_\tau \boldsymbol{T}^*(\boldsymbol{x}; \tau)}$$

must be the same constant for every $\boldsymbol{x}$.

Both conditions depend on the local density around $\boldsymbol{x}$ and generically fail, as different query points have different local weight distributions, so their bias-to-sensitivity ratios differ.

D.2 Effective Sample Size Cost

Even if one were to accept an approximate (non-uniform) bias reduction by lowering τ, there is a statistical cost. The effective sample size of the self-normalized estimator is

$$n_{\text{eff}} \;=\; \frac{n\,(\mathbb{E}[w])^2}{\mathbb{E}[w^2]} \;=\; \frac{n}{\lambda(\boldsymbol{x};\tau)},$$

where $\lambda = \mathbb{E}[w^2]/(\mathbb{E}[w])^2 \geq 1$. Writing $\beta := 1/\tau$ and $\phi(t) := \log \mathbb{E}[e^{-t\|\boldsymbol{x}-\boldsymbol{y}\|_2}]$ (convex in t), one checks $\log \lambda(\beta) = \phi(2\beta) - 2\phi(\beta)$, whose derivative $2[\phi'(2\beta) - \phi'(\beta)] \geq 0$ so $\log \lambda$ is monotonically increasing in $1/\tau$. Lowering τ therefore always reduces n_{eff}, concentrating the weights on fewer samples and increasing variance. In contrast, ABC removes the leading-order bias for a *fixed* τ without altering the weights or sacrificing sample efficiency.

D.3 Summary

Temperature adjustment fails to substitute for bias correction because:

1. the compensation equation (24) requires a global alignment that generically fails across query points;
2. lowering τ reduces effective sample size, increasing variance.

ABC avoids both issues: it corrects for the fixed target $\boldsymbol{T}^*(\boldsymbol{x};\tau)$, operates per-query-point via the local weight distribution, and is derived from the bias formula rather than a heuristic parameter adjustment.

E Training Algorithm

To better convey the overall training process with ABC, we provide pseudocode in Algorithm 1.

Algorithm 1 Drifting model training with ABC

Require: dataset $\mathcal{D}$, one-step generator g_θ, feature encoder ϕ, per-class batch size n, temperature τ

repeat
  Sample a class c, draw n real images I^+ of class c and n noise draws z, and set $I \leftarrow g_\theta(z, c)$
  $y^+ \leftarrow \phi(I^+)$, $x \leftarrow \phi(I)$ ▷ negatives y^- of x are the other generated features
  $\alpha^+ \leftarrow \mathrm{softmax}\big(-\|x - y^+\|_2/\tau\big)$, $\alpha^- \leftarrow \mathrm{softmax}\big(-\|x - y^-\|_2/\tau\big)$
  $T^+ \leftarrow \sum_i \alpha_i^+ y_i^+$, $T^- \leftarrow \sum_i \alpha_i^- y_i^-$ ▷ minibatch centroids
  $T^+ \leftarrow \big(1 - \sum_i (\alpha_i^+)^2\big) T^+ + \sum_i (\alpha_i^+)^2 y_i^+$ ▷ ABC
  $T^- \leftarrow \big(1 - \sum_i (\alpha_i^-)^2\big) T^- + \sum_i (\alpha_i^-)^2 y_i^-$ ▷ ABC
  $V \leftarrow T^+ - T^-$ ▷ drifting field at x
  $\mathcal{L} \leftarrow \big\|x - \mathrm{sg}[x + V]\big\|^2$ ▷ sg: stop gradient
  Update θ with $\nabla_\theta \mathcal{L}$
until converged

F Toy Experiment Details

Data distribution. The reference distribution p is a four-mode isotropic Gaussian in $\mathbb{R}^2$ with centers at $(\pm 0.5, \pm 0.5)$ and per-mode standard deviation $\sigma = 0.1$. Each mode is selected with equal probability $1/4$. We draw a reference pool of $N = 10^6$ points from p.

Query points. We sample 200 query points $\boldsymbol{x}$ from the same distribution p (so queries lie near the data modes, mimicking the realistic setting where generated samples are close to data clusters).

Estimation protocol. For each sample size $n \in \{4, 8, 16, 32, 64, 128, 256, 512\}$ and each query $\boldsymbol{x}$, we repeat 200,000 independent trials. In each trial, we draw n points from the pool (with replacement), compute both the standard centroid $\boldsymbol{T_n}$ and the ABC-corrected centroid $\boldsymbol{T_n^{\mathrm{ABC}}}$, and accumulate

the trial-averaged centroids. The bias is then measured as $\|\bar{T}_n - \boldsymbol{T}^*\|$, where $\boldsymbol{T}^*$ is the exact target centroid computed from all $N = 10^6$ pool points.

Bias-corrected norm estimator. Since $\|\mathbb{E}[\boldsymbol{T}_n] - \boldsymbol{T}^*\|$ is not directly observable (we observe $\|\bar{T}_n - \boldsymbol{T}^*\|$, which is inflated by Monte Carlo noise), we apply a standard variance-corrected norm estimator: $\hat{b}^2 = \|\bar{T}_n - \boldsymbol{T}^*\|^2 - \mathrm{tr}(\hat{\Sigma})/M$, where $\hat{\Sigma}$ is the sample covariance of the $M = 200{,}000$ trial centroids. We report $\hat{b} = \sqrt{\max(0, \hat{b}^2)}$. The plotted curve in Figure 2 is then the arithmetic mean of the per-query $\hat{b}$ values over all 200 sampled queries at each n. Note that $\hat{b}$ is computed pointwise in $\boldsymbol{x}$, so the across-query aggregation is a diagnostic average rather than a uniform-in-$\boldsymbol{x}$ statement.

Temperatures. We run the experiment at $\tau \in \{0.1, 0.2\}$ to verify that the $O(1/n)$ and $O(1/n^2)$ scaling holds across different kernel bandwidths. All computations use float64 accumulators on GPU for numerical stability.

G Comparison with Classical Bias-Reduction Baselines

Beyond ABC, several existing methods target the leading $O(1/n)$ ratio bias of the standard SNIS centroid: the classical jackknife [23], a bootstrap correction, and the more recent BR-SNIS estimator [5]. We compare these four corrections, alongside the uncorrected Standard estimator, on the same toy setup as Section 5.1.

Baselines. Jackknife, Bootstrap, and ABC correct the standard SNIS centroid $\boldsymbol{T}_n = \sum_i \alpha_i \boldsymbol{y}_i$ on a fixed minibatch of n samples through different mechanisms, whereas BR-SNIS replaces $\boldsymbol{T}_n$ with an i-SIR Markov-chain estimator whose initialization bias decays with burn-in. All four target the leading $O(1/n)$ bias of $\boldsymbol{T}_n$.

Jackknife [23]: first-order bias correction via leave-one-out recomputation. The key observation is that if the bias of $\boldsymbol{T}_n$ is $c/n + O(1/n^2)$, the specific linear combination $\boldsymbol{T}_n^{\mathrm{jack}} = n\,\boldsymbol{T}_n - (n-1)\,\overline{\boldsymbol{T}_{n,-i}}$, where $\boldsymbol{T}_{n,-i}$ is the SNIS centroid with the i-th sample removed, cancels the leading $1/n$ term and leaves an $O(1/n^2)$ residual. For SNIS specifically the softmax-marginalization identity $\boldsymbol{T}_{n,-i} = (\boldsymbol{T}_n - \alpha_i \boldsymbol{y}_i)/(1-\alpha_i)$ lets us compute all n leave-one-out estimates at $O(nD)$ cost, though with a substantially larger constant than the standard estimator.

Bootstrap: Efron-style bias estimation via $B = 100$ bootstrap resamples. For each draw of n samples we resample B times with replacement, compute an SNIS estimate $\boldsymbol{T}_n^{*(b)}$ on each replicate, and approximate the bias by $\overline{\boldsymbol{T}_n^{*(b)}} - \boldsymbol{T}_n$; the corrected estimator is $\boldsymbol{T}_n^{\mathrm{boot}} = 2\,\boldsymbol{T}_n - \overline{\boldsymbol{T}_n^{*(b)}}$. Cost: $O(BnD)$ per estimate.

BR-SNIS [5]: iterated Sampling-Importance Resampling (i-SIR). Each iteration concatenates the current state with $n-1$ fresh reference samples to form a candidate pool, selects the next state with probability proportional to the SNIS weights, and accumulates the SNIS centroid; the final estimator is the average of the SNIS centroids across $K - k_0$ post-burn-in iterations. Viewed as a Markov-chain procedure, the chain mixes to the target distribution and the bias decays exponentially in the burn-in k_0. We use $K = 10$ and $k_0 = 1$. Cost: $O(KnD)$ per estimate, but requires $K(n-1)+1$ fresh reference samples per estimate, a substantially larger sample budget than the other methods, which operate on a fixed n-sample pool.

ABC (ours): in contrast to the three baselines above, which rely on resampling or iterative procedures, ABC uses the analytical leading-bias formula from Theorem 3.3 and substitutes minibatch averages for the unknown full-distribution expectations, yielding a closed-form correction $\boldsymbol{T}_n^{\mathbf{ABC}} = (1 - \sum_i \alpha_i^2)\,\boldsymbol{T}_n + \sum_i \alpha_i^2 \boldsymbol{y}_i$ at $O(nD)$ cost (a single additional matmul, same order as the baseline centroid), with no extra samples or resamples required.

Protocol. We use a CPU-scale variant of the Appendix F setup at $\tau = 0.1$. The reference pool contains 5×10^4 points drawn from the four-mode Gaussian, and we evaluate on 200 Gaussian query points ($\mathcal{N}(0, 0.25\, I_2)$). For each $n \in \{4, 8, 16, 32, 64, 128, 256\}$ we run 5,000 Monte Carlo trials per query, drawing n samples without replacement from the pool each trial, and report the per-query bias

Table 5: Bias, variance, and wall time at $n \in \{8, 32, 128\}$ ($\tau = 0.1$, 5,000 trials). BR-SNIS uses $K(n-1)+1$ fresh samples per estimate (with $K = 10$); the other methods use n.

n	Method	$\|\text{bias}\|$ (↓)	Var (↓)	Time, µs (↓)
8	Standard	6.24×10^{-2}	7.54×10^{-2}	**5.9**
	Jackknife	7.02×10^{-2}	1.95×10^{-1}	82.9
	Bootstrap	3.27×10^{-2}	1.08×10^{-1}	996.6
	BR-SNIS (i-SIR)	$\mathbf{7.89 \times 10^{-3}}$	$\mathbf{6.68 \times 10^{-3}}$	1527.6
	ABC (ours)	4.99×10^{-2}	8.03×10^{-2}	25.5
32	Standard	1.33×10^{-2}	8.07×10^{-3}	**11.2**
	Jackknife	3.40×10^{-3}	9.85×10^{-3}	136.0
	Bootstrap	7.07×10^{-3}	9.00×10^{-3}	1985.1
	BR-SNIS (i-SIR)	$\mathbf{1.09 \times 10^{-3}}$	$\mathbf{1.24 \times 10^{-3}}$	3521.9
	ABC (ours)	7.26×10^{-3}	8.61×10^{-3}	36.9
128	Standard	3.63×10^{-3}	1.99×10^{-3}	**21.3**
	Jackknife	6.20×10^{-4}	2.15×10^{-3}	325.2
	Bootstrap	9.94×10^{-4}	2.12×10^{-3}	3669.7
	BR-SNIS (i-SIR)	$\mathbf{2.18 \times 10^{-4}}$	$\mathbf{2.60 \times 10^{-4}}$	10185.4
	ABC (ours)	1.03×10^{-3}	2.10×10^{-3}	57.0

norm $\|\mathbb{E}[\boldsymbol{T_n}] - \boldsymbol{T^*}\|$ and total variance $\sum_d \text{Var}(\boldsymbol{T_{n,d}})$ averaged across queries. Unlike Appendix F, here we use the naive norm $\|\bar{T}_n - \boldsymbol{T^*}\|$ rather than the variance-corrected estimator $\hat{b}$, so the reported bias values are slightly inflated by Monte Carlo noise (most noticeable for the smallest biases, e.g., BR-SNIS at large n). Per-estimate wall time is the mean across trials (excluding the first 10 warmup measurements), measured on CPU with PyTorch 2.3.

Findings. Among the fixed-n-pool methods, ABC attains bias comparable to Jackknife and Bootstrap with variance comparable to the Standard estimator's (which Jackknife notably inflates at small n) and running several times faster than either baseline, with no additional samples or resamples. BR-SNIS attains the smallest bias but at a $K\times$-larger sample budget and much higher wall time.

H Hyperparameters and Compute

Architecture. We use DriftDiT-Small from Deng et al. [7]: a 26.6M-parameter diffusion transformer with a frozen DINOv2 multi-scale encoder (four layers) and bidirectional kernel normalization. Input resolution is 32×32.

Training hyperparameters. Table 6 summarizes the training configuration. All hyperparameters except training duration follow Deng et al. [7]; training duration is scaled inversely with n so that every configuration sees the same total of 6.24M positive samples per class.

Table 6: Training hyperparameters.

Hyperparameter	Value
Optimizer	AdamW (β_1=0.9, β_2=0.95)
Weight decay	0.01
Learning rate	2×10^{-4}
Warmup	2 000 linear steps
EMA decay	0.999
Drift-field temperatures τ	$\{0.02,\ 0.05,\ 0.2\}$
Steps per epoch	195
Epochs (n=64/32/16/8)	500/1 000/2 000/4 000
Total positive samples per class seen	6.24M
Seeds	$\{42, 43, 44\}$

Evaluation. FID is computed on 50 000 generated samples per checkpoint. Evaluation frequency during training ranges from every 25 to every 200 epochs depending on configuration. The smoothed trajectories in Figure 3 use a rolling mean over $\{7, 5, 3, 3\}$ evaluation checkpoints for $n \in \{8, 16, 32, 64\}$ respectively (smaller-n panels run more epochs and are smoothed more aggressively), with bands showing mean $\pm$ max-deviation across the three seeds.

Hardware and software. All experiments run on a single NVIDIA A100 80 GB GPU with PyTorch and CUDA. Average per-epoch wall time is approximately 37.1 s for Standard and 42.8 s for ABC, with peak GPU memory of 4 729 vs 4 731 MB (Table 3). Total compute for the main CIFAR-10 grid (4 sample sizes $\times$ 2 methods $\times$ 3 seeds $=$ 24 runs) is approximately 700 A100-hours of single-GPU-equivalent time, plus additional time for ablations and toy experiments.

I Additional Results on CIFAR-100

As a consistency check on a harder dataset, we run single-seed ABC vs Standard on CIFAR-100 using the same DriftDiT-Small architecture and hyperparameters as Section 5.2, except for the training epochs, which are reduced because of the higher per-epoch cost on CIFAR-100. Compute parity with the CIFAR-10 budget is maintained for n=16 and n=32, while n=8 is budget-reduced and used only as a preliminary check. The results show that ABC also improves both last-5 average and best FID on CIFAR-100.

Table 7: FID on CIFAR-100.

n	Epochs	Last-5 average FID (↓)		Best FID (↓)	
		Standard	ABC	Standard	ABC
8	1 000	4.17	**4.05** ↓0.12	4.06	**3.74** ↓0.32
16	2 000	2.64	**2.52** ↓0.12	2.52	**2.40** ↓0.12
32	1 000	2.26	**2.17** ↓0.09	2.20	**2.10** ↓0.10

J Precision and Recall

To complement the FID metric used in the main text, we additionally report the improved precision and recall of Kynkäänniemi et al. [17] on CIFAR-10. For each run, we take the EMA checkpoint with the lowest FID, generate 50,000 samples with the same sampling configuration as the FID evaluation, and compute both metrics on InceptionV3 pool3 features against the CIFAR-10 training set. We report both k=3 (the setting of Kynkäänniemi et al. [17]) and k=5 (the setting in the usage example of the `prdc` library[1]). Table 8 reports the results.

Table 8: Improved precision and recall on CIFAR-10.

n	Recall, k=3 (↑)		Recall, k=5 (↑)		Precision, k=3 (↑)		Precision, k=5 (↑)	
	Standard	ABC	Standard	ABC	Standard	ABC	Standard	ABC
8	0.26	**0.33** ↑0.07	0.36	**0.44** ↑0.08	0.64	**0.65** ↑0.01	0.75	0.75
16	0.39	**0.41** ↑0.02	0.50	**0.53** ↑0.03	**0.67**	0.66 ↓0.01	**0.77**	0.76 ↓0.01
32	0.47	**0.49** ↑0.02	0.58	**0.60** ↑0.02	**0.66**	0.65 ↓0.01	**0.76**	0.75 ↓0.01
64	0.48	0.48	0.59	0.59	0.66	0.66	0.76	0.76

It is clear that ABC improves recall rather than precision. This is consistent with what our theory predicts. Theorem 3.3 shows the bias is a contraction of the centroids of the drifting field, which suppresses diversity, and ABC removes precisely this contraction. The correction is therefore expected to recover recall and leave precision untouched, exactly as observed. Also, the recall gain from ABC is large at n=8, shrinks through n=16 and n=32, and vanishes at n=64, which is consistent with the $O(1/n)$ trend. These results provide an independent signal beyond FID that the reported improvements reflect the predicted mechanism of bias correction.

[1] `https://github.com/clovaai/generative-evaluation-prdc`

K Shrinkage-Only Ablation

To further confirm that the gains come from bias correction, we run an ablation that separates the bias correction from the shrinkage factor. We implement a shrinkage-only variant that applies the damping factor while dropping the corrective term, i.e., $\boldsymbol{T}_{\boldsymbol{n}}^{\mathbf{shrink}} = (1 - \sum_i \alpha_i^2)\, \boldsymbol{T}_{\boldsymbol{n}}$, and report the mean FID trend during training over three seeds in Figure 5.

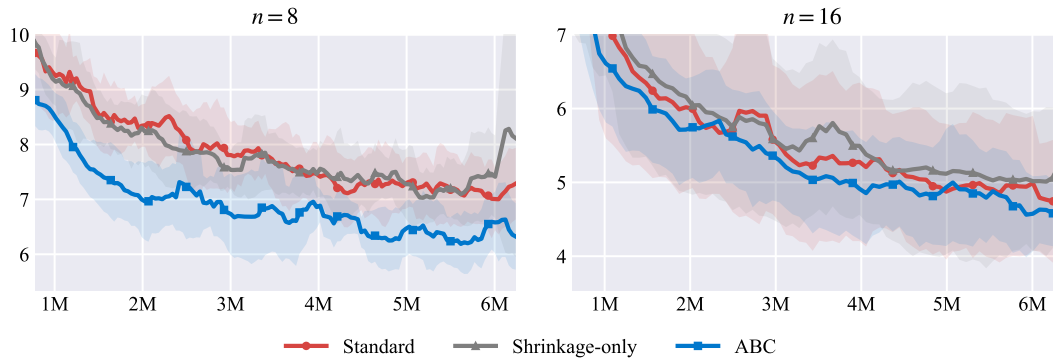


Figure 5: FID trajectories at n=8 and n=16 for Standard, shrinkage-only, and ABC. Bands show mean $\pm$ max-deviation across three seeds.

These results match the theoretical prediction. The corrective term that the shrinkage-only variant drops is itself of order $O(1/n)$, so its residual bias remains $O(1/n)$, the same order as Standard. Consistently, the shrinkage-only variant stays close to or even worse than Standard at both n, and is clearly behind ABC throughout. This supports the conclusion that ABC's improvement comes from removing the bias rather than from shrinkage or variance damping.

L Experiments with n=128

We additionally train Standard and ABC on CIFAR-10 at n=128 with a single seed, using the same total optimizer steps as the n=64 runs (500 epochs). Over the last five evaluations, the average FID is 3.08 for Standard and 2.92 for ABC, and the best FID is 2.88 and 2.68, respectively. The results are consistent with the $O(1/n)$ bias, while ABC still retains a small advantage.

M Limitations and Future Work

The primary contribution of this paper is the theoretical analysis of the $O(1/n)$ subsampling bias in drifting models and the analytical correction (ABC) that addresses it. Our experiments validate this theory rather than pursue state-of-the-art generation quality. Compute constraints also limit the experimental scope. Follow-up and application-oriented work could further tune the temperature τ and other hyperparameters in the presence of ABC to achieve better task-specific performance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions stated in the abstract and Section 1 are supported by the theory in Sections 3 and 4 and the experiments in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix M. The assumptions of the theory are discussed in Remark 3.4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Assumptions are stated in each theorem and proposition, and full proofs are in Appendices B and C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setup is described in Section 5.2 and Appendix H, the implementation of ABC in Section 4.3 and Appendix E, and the toy setup in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is available at `https://github.com/lan-qing/DriftingABC`, and all datasets used are public.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5.2 and Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main results report mean $\pm$ max-deviation across three random seeds (Section 5.2).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper corrects an estimator bias in existing generative models and has no direct societal impact beyond that of generative modeling in general.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases no data or models with a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the drifting models [7], DINOv2 [21], the CIFAR datasets [16], and the evaluation metrics [12, 17], and use them under their respective licenses for research purposes only.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code is documented in the README of its repository.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core methodology of this paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.