Compute-Optimal Training and Sampling Schedules for Diffusion Models via Spectral Kernel Error Budgets

Liz Lemma Future Detective

January 20, 2026

Abstract

Modern diffusion systems (2026) are dominated by compute tradeoffs: how to allocate training effort across noise levels, how many sampling steps to take, and where to stop. Building on fine-grained Wasserstein error expansions in the Gaussian-linear setting, we formalize diffusion training+sampling as a single budgeted optimization problem. We derive a kernelized surrogate for the end-to-end expected W_2^2 error that isolates (i) truncation/early-stopping bias, (ii) discretization bias, and (iii) a kernel-weighted accumulation of score estimation error whose coefficients depend on the data power spectrum and the diffusion/solver schedule. Under a stylized but analyzable model of training noise—capturing both finite-data effects and constant-step optimizer stationarity—we solve for compute-optimal training weights $w^*(t)$ (or update allocations $m^*(t)$), sampler step schedules $\gamma^*(t)$, and optimal stopping times r^*. The optimum is characterized by equalized marginal improvements, yielding explicit scaling laws that generalize prior heuristics (e.g., r^* scaling with optimizer noise and schedule smoothness). We provide matching lower bounds in the Gaussian-linear model showing that, up to constants (and inevitable log factors for VP schedules), no method can beat the derived compute–accuracy frontier. We outline an implementable scheduler that estimates required spectral quantities online and validates the theory on synthetic anisotropic Gaussians and medium-scale diffusion models.

Table of Contents

1. 1. Introduction: compute is the bottleneck; why schedule co-design matters; summary of kernelized error budgets and main theorems.
2. 2. Background and source connection: DSM/EDM training, reverse SDE / probability-flow ODE, Wasserstein/Bures for Gaussians; recap of kernel-norm error forms in Gaussian-linear analysis.

3. 3. Gaussian-linear diffusion as an error budget: define $E^{(0)}(r)$, $E^{\mathrm{disc}}(r, \gamma)$, and kernel-weighted score-error terms; reduce to spectral sums/integrals over eigenvalues $\{\lambda_i\}$.

4. 4. Training noise model under resource allocation: define $m(t)$ and derive/assume a generic covariance law $\mathrm{Cov}(\theta_t - \theta_t^*) \preceq \frac{A(t)}{m(t)} + \tau B(t)$; discuss how $A(t), B(t)$ arise from (finite data, constant-step optimizer) and how to estimate them.

5. 5. The budgeted co-design problem: formal statement; discretized-time and continuous-time versions; convexity conditions; feasibility and stability constraints for sampling.

6. 6. Optimal training allocation $m^*(t)$: KKT conditions; closed-form solutions (water-filling / square-root rules) under common cost models; interpretation as equal marginal gain weighted by kernel sensitivity.

7. 7. Optimal sampling schedule $\gamma^*(t)$ and stopping time r^*: derive optimality conditions under a sampling compute budget; provide explicit scalings for VP/VE/EDM schedule families; discuss solver order q and discretization decay.

8. 8. Matching lower bounds in the Gaussian-linear model: minimax lower bounds for score estimation under compute constraints; reduction to linear regression/mean estimation; propagate to a lower bound on achievable W_2^2; compare to achieved upper bound (gap and logs).

9. 9. Practical scheduler: how to estimate spectrum/bandpower and training error statistics; implementable algorithm that outputs $w(t)$, $\gamma(t)$, and r; robustness considerations.

10. 10. Experiments (recommended): synthetic anisotropic Gaussians to verify tightness; ablations on training reweighting vs sampling steps; medium-scale diffusion/EDM models to test predictive power of the schedule; show compute–quality frontier shifts as predicted.

11. 11. Limitations and extensions: correlated-time errors (shared network across t); non-Gaussian data via local linearization; connections to distillation and solver selection.

12. 12. Conclusion: kernelized compute-optimal co-design as a unifying perspective; open problems.

1 Introduction

We take the point of view that, in contemporary diffusion models, the limiting resource is not representational capacity but compute: the number of stochastic gradient evaluations available for training and the number of function evaluations available for sampling. In the regime where the score network is sufficiently expressive and optimization is run long enough to approach a stationary stochastic dynamics, improvements in sample quality are predominantly controlled by how this compute is distributed across noise levels during training and across time steps during sampling. Consequently, it is not natural to treat "training schedule" (noise reweighting, curriculum, or per-time sampling probability) and "sampling schedule" (step sizes, solver order, and early stopping) as independent design problems. We instead view them as a single constrained optimization problem: under a fixed total budget, one must decide *where* to reduce score error (training allocation) and *where* to spend steps to reduce numerical and truncation error (sampling allocation).

Our analysis is organized around a kernelized error budget for the end-to-end generative discrepancy, measured in squared Wasserstein distance. The guiding principle is that, after linearization in the score error, the effect of imperfect score estimation at a given noise level is not uniform: it is amplified or damped by the sampler dynamics and by the data covariance geometry. This yields an explicit sensitivity operator, denoted $K(r, t; C)$, which maps score-estimation uncertainty at time t into contribution to the final sampling error when the reverse-time procedure is stopped at terminal reverse-time r. The introduction of K is not merely notational. It allows us to convert a high-dimensional, time-inhomogeneous learning problem into an analytically tractable resource-allocation objective whose structure is essentially that of a weighted integral of variances.

The main technical advantage of working in the Gaussian-linear setting is that the relevant objects admit closed forms in the eigenbasis of the data covariance C. In particular, both (i) the Wasserstein geometry for Gaussian laws (which reduces to mean error plus a Bures-type term) and (ii) the reverse diffusion dynamics with linear scores lead to expressions where each spectral direction u_i can be tracked independently, up to kernel couplings that can be summarized as spectral kernels $k_r(\lambda_i, \lambda_j)$. This makes it possible to state and solve a co-design problem that depends on the data distribution only through its power spectrum $\{\lambda_i\}$ (or bandpower approximations), and on the diffusion process only through its schedule-dependent coefficients.

The first theorem establishes the form of the compute–accuracy decomposition. The surrogate expected error splits into three contributions: a truncation (or early-stopping) term $E^{(0)}(r)$, a discretization term $E^{\mathrm{disc}}(r, \gamma)$ determined by the solver family and step sizes, and a kernel-weighted score-error term. The latter takes the form of an integral over noise levels of the step size $\gamma(t)$ times an inner product $\langle K(r,t;C), \mathrm{Cov}(\theta_t - \theta_t^*)\rangle$. Under a stationary constant-step optimizer model, we bound the parameter covariance by a sum of a *reducible* component scaling as $1/m(t)$ and an *irreducible* component scaling as τ. This yields an explicit objective of the schematic form

$$E^{(0)}(r) + E^{\mathrm{disc}}(r,\gamma) + \int_r^T \gamma(t)\Big\langle K(r,t;C), \frac{A(t)}{m(t)} + \tau B(t)\Big\rangle\, dt,$$

up to higher-order remainders controlled by the linearization accuracy and the discretization granularity. The content of the theorem is not the existence of such a decomposition in the abstract, but rather that each term is computable from the diffusion schedule and the spectrum, and that the dependence on $m(t)$ is convex.

The second theorem concerns the optimal distribution of training updates across noise levels when the sampling schedule and stopping time are held fixed. The allocation subproblem is a continuum analogue of classical variance-reduction under cost constraints. Because the integrand is proportional to $1/m(t)$ with nonnegative weights, convexity is immediate and the KKT conditions yield a closed-form minimizer. In particular, on the active set where training is worthwhile, the optimal $m^*(t)$ scales as the square root of a "sensitivity over cost" ratio:

$$m^*(t) \;\propto\; \sqrt{\frac{\gamma(t)\langle K(r,t;C), A(t)\rangle}{c_{\mathrm{train}}(t)}}.$$

Equivalently, the marginal error decrease per unit training compute is equalized across noise levels. This rule provides a mathematically precise statement of an intuition that is often invoked informally: one should train more where errors matter more for sampling, but the correct notion of "matter" is determined jointly by the sampler kernel K, the spectrum of C, and the optimizer-induced noise model $A(t), B(t)$.

The third theorem addresses a distinct design choice: where to stop the reverse-time procedure. Decreasing r reduces truncation bias but typically increases accumulated error from irreducible training noise (and, depending on the schedule, may also stress discretization). Under regularity assumptions on near-zero noise behavior, we obtain an explicit scaling law for the compute-optimal r^* by balancing the leading truncation term against the dominant optimizer-limited accumulation term. The resulting expression explains, in a unified way, why VP-like schedules can exhibit logarithmic sensitivity to the smallest noise level and why early stopping is not merely a numerical convenience but can be compute-optimal.

Finally, the fourth theorem provides a matching-order lower bound in the same Gaussian-linear model. It shows that the dependence on $m(t)$ and τ in the kernelized term is not an artifact of the proof technique: any method that accesses training through at most $m(t)$ stochastic gradients at noise level t must incur error of order $1/m(t)$ in the relevant directions, and any constant-step stationary optimizer induces an unavoidable $\Omega(\tau)$ variance floor. When propagated through the sampler sensitivity operator, these constraints yield a minimax lower bound that matches the upper bound achieved by the KKT-optimal allocation, up to constants and the discretization order. Thus, within the assumed regime, the derived schedules are not only principled but essentially optimal.

The overall consequence is a concrete co-design methodology: estimate (or bandpower-approximate) the spectrum of C, specify a diffusion schedule and solver family, posit or fit a training-noise model $A(t), B(t)$ together with cost models, and then compute $m^*(t)$, $\gamma(t)$, and r^* by solving a small collection of convex subproblems and a one-dimensional search over candidate stopping times. The subsequent sections supply the background connecting DSM/EDM training and reverse-time solvers to the Gaussian Wasserstein/Bures geometry, and then derive the kernel representations that make this optimization explicit.

2 Background: DSM/EDM training, reverse-time dynamics, and Gaussian Wasserstein geometry

We recall the standard correspondence between (i) denoising-score training at fixed noise level and (ii) reverse-time sampling driven by the learned score. In the unified VP/VE/EDM parameterization, we may represent the forward noising mechanism at time $t \in [0, T]$ by

$$x_t \;=\; s_t x_0 + \sigma_t z, \qquad z \sim \mathcal{N}(0, I_d), \quad x_0 \sim p_{\mathrm{data}},$$

where s_t and σ_t are schedule-dependent scalars (for VP, s_t decays and σ_t grows; for VE, $s_t \equiv 1$ and σ_t grows). When $p_{\mathrm{data}} = \mathcal{N}(\mu, C)$ with $C \succ 0$, the marginal law is explicit:

$$p_t \;=\; \mathcal{N}\big(s_t\mu,\; \Sigma_t\big), \qquad \Sigma_t := s_t^2 C + \sigma_t^2 I_d,$$

and hence the exact score is linear,

$$s^*(x,t) \;:=\; \nabla_x \log p_t(x) \;=\; -\Sigma_t^{-1}(x - s_t\mu).$$

This linearity is the basic reason the Gaussian setting admits closed-form error propagation: any score model which is linear in x at each t can represent $s^*(\cdot, t)$ exactly, and deviations from optimality can be summarized by time-dependent parameter errors.

At the level of training, DSM/EDM objectives fit the score by a weighted regression against either the perturbation z or the conditional score. Concretely, for a chosen weight $w(t)$ (or, operationally, a sampling frequency over t), one considers objectives of the schematic form

$$\min_{\theta} \int_0^T w(t)\, \mathbb{E}\big[\|s_\theta(x_t,t) - s^*(x_t,t)\|^2\big]\, dt,$$

possibly up to schedule-dependent rescalings (e.g. EDM parameterizations that predict x_0 or ϵ can be rewritten as score matching by an invertible linear transform in the Gaussian case). Under the assumption that, for each fixed t, optimization runs long enough to reach a stationary regime with constant optimizer step size τ, the residual parameter error $\theta_t - \theta_t^*$ behaves like a noisy equilibrium whose covariance decomposes into a part reducible by more updates and a part induced by stationary optimizer noise. We encode this through a bound of the form $\mathrm{Cov}(\theta_t - \theta_t^*) \preceq A(t)/m(t) + \tau B(t)$, where $m(t)$ is the effective number of gradient evaluations allocated to noise level t (or to the time bin containing t). In the Gaussian-linear regime this is not merely a qualitative statement: since the population DSM objective is quadratic, the optimum θ_t^* is unique in the identifiable subspace and the covariance scaling in $1/m(t)$ corresponds to the familiar variance decrease under repeated stochastic estimation, while $\tau B(t)$ represents the non-vanishing stationary variance floor.

On the sampling side, the forward diffusion is represented as an SDE

$$dx_t \;=\; f(x_t,t)\, dt + g(t)\, dW_t$$

(for appropriate drift f and diffusion coefficient g determined by the schedule). The reverse-time SDE, run from $t = T$ down to $t = r$ (equivalently, reverse-time variable $r = T - t$ increasing), takes the standard form

$$dx_t \;=\; \big(f(x_t,t) - g(t)^2\, \nabla_x \log p_t(x_t)\big)\, dt + g(t)\, d\bar{W}_t,$$

and replacing $\nabla \log p_t$ by the learned score $s_\theta(\cdot,t)$ yields the practical sampling dynamics. For the probability-flow ODE, the stochastic term is removed and the reverse drift becomes $f - \frac{1}{2}g^2\nabla \log p_t$; both cases share the key structural property used later: the dependence on the score enters linearly in the drift. Discretization with step sizes $\{\gamma_k\}$ (or a piecewise-constant $\gamma(t)$ on a time grid) produces a numerical method whose global error decomposes into (i) truncation effects from stopping at $t = r$ rather than $t = 0$ and (ii) solver-dependent discretization effects, both of which may be analyzed separately from score-estimation error once we work to leading order in perturbations.

The metric used to assess end-to-end generative quality is the squared Wasserstein distance. For Gaussian laws $P = \mathcal{N}(m_1, \Sigma_1)$ and $Q = \mathcal{N}(m_2, \Sigma_2)$,

$$W_2^2(P,Q) \;=\; \|m_1 - m_2\|^2 + \mathrm{Tr}\Big(\Sigma_1 + \Sigma_2 - 2\big(\Sigma_2^{1/2}\Sigma_1\Sigma_2^{1/2}\big)^{1/2}\Big),$$

where the second term is the Bures metric between covariances. In perturbative regimes, the Bures term admits a quadratic expansion around a reference covariance: if $\Sigma = \Sigma_0 + \Delta$ with $\|\Delta\|$ small, then $W_2^2(\mathcal{N}(0,\Sigma), \mathcal{N}(0,\Sigma_0))$ is, to second order, a positive semidefinite quadratic form in Δ whose coefficients are diagonal in the eigenbasis of Σ_0 (equivalently, of C after propagating through the linear forward map). In particular, if $Cu_i = \lambda_i u_i$, then quadratic forms associated with Bures-type expansions decompose into sums over pairs (i,j) with weights depending smoothly on (λ_i, λ_j), a fact we will exploit by introducing spectral kernels.

Finally, we recall how these components combine in the Gaussian-linear analysis. Because the reverse dynamics is linear in x when the score is linear, the effect of a score error $e_t(x) := s_\theta(x,t) - s^*(x,t)$ can be propagated through the solver as a linear response: the induced perturbations in the terminal mean and covariance are linear functionals of the time-indexed error process, with coefficients determined by the diffusion schedule and the discretization scheme. Consequently, the leading contribution to $\mathbb{E}[W_2^2]$ from score error is quadratic in e_t; after taking expectation over training randomness and over the sampling path, this quadratic term becomes an inner product between a sampler-induced sensitivity operator and the covariance of the score parameters. This is the origin of the kernel representation $K(r,t;C)$: it summarizes, at each noise level t, how uncertainty in the learned linear score (encoded by $\mathrm{Cov}(\theta_t - \theta_t^*)$) is amplified by the remaining reverse-time evolution down to the stopping time r. In the eigenbasis of C, $K(r,t;C)$ reduces to explicit spectral weights $k_r(\lambda_i, \lambda_j)$, so that both training and sampling design can be expressed in terms of the power spectrum $\{\lambda_i\}$ (or its band-power approximation) together with schedule-dependent scalar coefficients. This background will allow us, in the next section, to write the sampling error as an error budget with a truncation term, a discretization term, and a kernel-weighted score-error term amenable to resource allocation.

3 Gaussian-linear diffusion as an error budget

We now specialize the end-to-end sampling error to the Gaussian-linear regime and isolate the three contributions that will later be traded off by schedule design: truncation from stopping at $r > 0$, discretization from the numerical solver, and amplification of training-induced score error through the remaining reverse-time dynamics. Throughout we regard the sampler as producing an output at terminal forward-time $t = r$ (reverse-time $T - r$), followed by the standard deterministic "denoising" rescaling $\Pi_r(x) := x/s_r$ in the unified VP/VE parameterization, so that the output is comparable to p_{data} even when $r > 0$. (In VE one has $s_r \equiv 1$ and Π_r is the identity.)

Truncation / early-stopping bias. Let q_r^* denote the law obtained by running the *exact* reverse-time dynamics (SDE or probability-flow ODE, as chosen) with the *exact* score and with vanishing discretization error, but stopped at time r and postprocessed by Π_r. In the Gaussian case this law is explicit. Indeed, if $x_r \sim p_r = \mathcal{N}(s_r\mu,\, s_r^2 C + \sigma_r^2 I)$ then $\Pi_r(x_r)$ is Gaussian with mean μ and covariance

$$\widetilde{C}_r \;:=\; C + \delta_r^2 I_d, \qquad \delta_r^2 := \left(\frac{\sigma_r}{s_r}\right)^2,$$

so we define the truncation bias term by

$$E^{(0)}(r) \;:=\; W_2^2\big(\mathcal{N}(\mu, C),\, \mathcal{N}(\mu, \widetilde{C}_r)\big).$$

Since C and $\widetilde{C}_r$ commute, the Bures term diagonalizes in the eigenbasis $\{u_i\}$ of C, yielding the closed form

$$E^{(0)}(r) \;=\; \sum_{i=1}^{d} \left(\sqrt{\lambda_i + \delta_r^2} - \sqrt{\lambda_i}\right)^2.$$

This is the irreducible price of stopping at r even with an oracle score and an exact integrator; it vanishes as $r \downarrow 0$ under any schedule for which $\delta_r \to 0$.

Discretization bias. Fix a solver family of order q (Euler, Heun, Runge–Kutta, etc.) and a reverse-time grid $T = t_0 > t_1 > \cdots > t_K = r$ with step sizes $\gamma_k := t_{k-1} - t_k$ (equivalently a piecewise-constant step-size proxy $\gamma(t)$). Let $q_{r,\gamma}^*$ denote the law of the discretized sampler when driven by the *exact* score. We define the discretization term as the residual error relative to q_r^*:

$$E^{\mathrm{disc}}(r,\gamma) \;:=\; W_2^2\big(q_r^*,\, q_{r,\gamma}^*\big).$$

In the Gaussian-linear setting the reverse dynamics is affine, hence both q_r^* and $q_{r,\gamma}^*$ are Gaussian and E^{disc} is again a function of mean/covariance discrepancies. While the exact closed form depends on the chosen solver and on whether we sample the SDE or the probability-flow ODE, in all cases we may view $E^{\mathrm{disc}}(r,\gamma)$ as a deterministic functional of the schedule and the grid satisfying the expected order condition

$$E^{\mathrm{disc}}(r,\gamma) \;=\; O\Big(\sum_{k=1}^{K} \gamma_k^{2q}\Big)$$

under standard stability regularity (bounded coefficients and a step-size feasibility condition depending on $\lambda_{\max}$ through the linear drift). We will keep E^{disc} as an explicit term in the surrogate rather than absorbing it into higher-order remainders, since it competes directly with the training-driven term when sampling compute is limited.

Kernel-weighted score-error term. We next quantify how imperfect scores perturb the terminal output. At each time t, let the learned score be $s_\theta(\cdot, t)$ and define the score error process $e_t(x) := s_\theta(x,t) - s^*(x,t)$. In the Gaussian-linear regime e_t is linear in x (and affine if $\mu \neq 0$), hence its effect on the sampler can be expressed via linear response: the perturbations in the terminal mean and covariance are linear functionals of the time-indexed parameter errors. Consequently, after expanding W_2^2 to second order around the oracle trajectory and taking expectation over training randomness (and, for the SDE, the sampling noise), the leading contribution from score error is quadratic in the parameter error and therefore depends on its covariance.

We encode this dependence by a time-dependent positive semidefinite operator $K(r,t;C)$ such that the leading score-error contribution takes the form

$$E^{\text{score}}(r,\gamma,m) \;:=\; \int_r^T \gamma(t)\,\Big\langle K(r,t;C),\, \operatorname{Cov}\big(\theta_t - \theta_t^*\big)\Big\rangle\, dt,$$

where $\langle\cdot,\cdot\rangle$ denotes the Frobenius inner product after identifying parameters with the corresponding linear-score coefficients (e.g. the matrix multiplying x in $s_\theta(x,t)$, together with the affine part when present). The factor $\gamma(t)$ reflects that, on a grid, each step contributes proportionally to its local integration length, and in the continuous-time limit the sum becomes an integral.

The crucial point is that $K(r,t;C)$ is explicit in the eigenbasis of C. Writing $\Sigma_t = s_t^2 C + \sigma_t^2 I$ so that $\Sigma_t u_i = (s_t^2\lambda_i + \sigma_t^2)u_i$, the reverse-time propagator from t down to r acts diagonally on each eigen-direction. As a result, the second-order Bures expansion for the terminal covariance perturbation decomposes into pairwise interactions between eigen-directions, and we may write

$$\Big\langle K(r,t;C),\, \operatorname{Cov}\big(\theta_t - \theta_t^*\big)\Big\rangle \;=\; \sum_{i=1}^{d}\sum_{j=1}^{d} k_{r,t}(\lambda_i,\lambda_j)\;\Xi_t(i,j),$$

where $k_{r,t}(\lambda_i,\lambda_j) \geq 0$ is a schedule- and solver-dependent spectral kernel (the "sampler sensitivity") and $\Xi_t(i,j)$ denotes the appropriate covariance component of the score-parameter error in the (u_i,u_j) block. In particular, when the learned linear score (and its estimation noise) respects the eigenspace decomposition of C—a common situation in isotropic parameterizations or when one analyzes each eigen-direction separately—the off-diagonal components vanish and the score term reduces to a one-dimensional spectral sum $\sum_i \kappa_{r,t}(\lambda_i)\operatorname{Var}(\theta_{t,i} - \theta_{t,i}^*)$ for an explicit $\kappa_{r,t}$.

Spectral/bandpower reduction. For large d it is convenient to replace $\{\lambda_i\}$ by a spectral measure $\nu := \frac{1}{d}\sum_{i=1}^d \delta_{\lambda_i}$ or by a bandpower approximation $\{\bar\lambda_b, n_b\}_{b=1}^B$ (band representative and multiplicity). Since $k_{r,t}(\cdot,\cdot)$ is a smooth function of its arguments in the Gaussian-linear formulas, we may approximate the double sum by

$$\sum_{i,j} k_{r,t}(\lambda_i, \lambda_j)\, \Xi_t(i,j) \;\approx\; \sum_{b=1}^{B} \sum_{b'=1}^{B} n_b n_{b'}\, k_{r,t}(\bar{\lambda}_b, \bar{\lambda}_{b'})\, \bar{\Xi}_t(b,b'),$$

or, in the continuum limit, by a kernel integral $\iint k_{r,t}(\lambda, \lambda')\, d\nu(\lambda)\, d\nu(\lambda')$ with an analogous averaged covariance term. This reduction is the step that makes co-design computationally tractable: all dependence on the data distribution enters through its spectrum, and all dependence on the sampler enters through $k_{r,t}$ and the discretization functional E^{disc}.

Collecting the preceding pieces, we arrive at the surrogate error budget used throughout:

$$\mathbb{E}\big[W_2^2(p_{\text{data}}, q_r)\big] \;\approx\; E^{(0)}(r) + E^{\text{disc}}(r,\gamma) + \int_r^T \gamma(t)\, \Big\langle K(r,t;C),\, \operatorname{Cov}\big(\theta_t - \theta_t^*\big) \Big\rangle\, dt,$$

with remainder terms of higher order in (score-error magnitude, discretization scale). The remaining task is to relate $\operatorname{Cov}(\theta_t - \theta_t^*)$ to training compute via a resource allocation model, which we do next.

4 Training noise model under resource allocation

We now connect the covariance term $\operatorname{Cov}(\theta_t - \theta_t^*)$ appearing in the kernelized error budget to the amount of training compute expended at each noise level. The co-design problem will treat training as an allocatable resource over $t \in [r,T]$: we may choose to spend more gradient evaluations at those noise levels whose score accuracy is amplified most strongly by the sampler kernel $K(r,t;C)$.

Allocating training updates across noise levels. We model training as producing, for each t, a score parameter vector (or matrix) θ_t by optimizing a time-conditioned denoising-score-matching objective. Concretely, when training proceeds by sampling noise levels and taking stochastic gradient steps, the *allocation* is described by a nonnegative function $m(t)$ such that, for any measurable set $U \subseteq [r,T]$, the quantity $\int_U m(t)\, dt$ is proportional to the expected number of gradient evaluations performed with noise levels in U. Equivalently, if one uses a normalized sampling distribution $w(t)$ over noise levels and a total of M training updates, then $m(t)$ plays the role $m(t) = M\, w(t)$ (in continuous time) or $m_i = M w_i$ (in a discretized binning). We emphasize that $m(t)$ is an *effective* number of updates: it can absorb minibatch size, gradient accumulation, or reuse of cached features, and it will later be paired with a cost density $c_{\text{train}}(t)$.

Stationary constant-step covariance: reducible versus irreducible. Our surrogate is driven by the second moment of the parameter error at each t. We adopt the standing assumption that, for each t, the optimizer is run at constant step size τ sufficiently long that the iterates are well approximated by a stationary distribution in a neighborhood of the time-t optimum θ_t^* (the DSM minimizer in the Gaussian-linear regime, or its NTK-linearized analogue). In this regime it is standard to approximate constant-step stochastic gradient methods by a linear stochastic recursion around θ_t^*, or by an Ornstein–Uhlenbeck diffusion after appropriate scaling. Both viewpoints yield a decomposition of the stationary covariance into (i) a component that decreases with the number of independent gradient samples used to form the estimate of θ_t^* and/or to reduce gradient noise, and (ii) a component that persists even as the number of updates grows, because constant-step methods do not converge to a point mass.

Motivated by these considerations, we posit the generic covariance upper bound

$$\mathrm{Cov}(\theta_t - \theta_t^*) \;\preceq\; \frac{A(t)}{m(t)} \;+\; \tau\, B(t), \tag{1}$$

where $A(t) \succeq 0$ and $B(t) \succeq 0$ are time-dependent matrices in the parameter space (matching the representation used by the kernel operator $K(r, t; C)$). The term $A(t)/m(t)$ is the *reducible* part: holding all else fixed, spending twice as many effective gradient evaluations at time t halves this contribution. The term $\tau B(t)$ is the *irreducible* optimizer-noise floor induced by constant-step stationarity: it vanishes only in the limit $\tau \downarrow 0$ or under variance-reduced / annealed-step procedures not modeled here.

Where do $A(t)$ and $B(t)$ come from? In the Gaussian-linear DSM setting, the time-t objective is a quadratic function of the linear score coefficients. Writing the population loss in the form

$$\mathcal{L}_t(\theta) \;=\; \frac{1}{2}\langle \theta - \theta_t^*,\, H_t(\theta - \theta_t^*)\rangle,$$

with Hessian $H_t \succeq 0$, a single stochastic gradient takes the form $g(\theta; z, t) = H_t(\theta - \theta_t^*) + \varepsilon_t(z)$, where z denotes the randomness from drawing a data point and noise, and ε_t is a zero-mean gradient noise with covariance $\Sigma_t := \mathbb{E}[\varepsilon_t \varepsilon_t^\top]$. If we replace $m(t)$ independent gradient samples by their average (or, equivalently, if we imagine a batch size scaling linearly with $m(t)$), then the effective noise covariance scales as $\Sigma_t/m(t)$, yielding parameter fluctuations on the order of $H_t^{-1}\Sigma_t H_t^{-1}/m(t)$. This motivates taking $A(t)$ proportional to $H_t^{-1}\Sigma_t H_t^{-1}$ (with additional factors depending on momentum/preconditioning and on the precise parameterization of the score). Separately, for constant-step SGD without averaging, the stationary covariance solves an approximate Lyapunov equation of the form

$$H_t\,\Pi_t + \Pi_t\,H_t \;\approx\; \tau\,\Sigma_t,$$

so that Π_t is of order τ; in the commutative or well-conditioned regime one obtains the heuristic $\Pi_t \approx \frac{\tau}{2} H_t^{-1} \Sigma_t H_t^{-1}$, motivating $B(t)$ as a (possibly preconditioned) analogue of $H_t^{-1} \Sigma_t H_t^{-1}$. In more realistic settings with finite dataset size N, the gradient noise itself decomposes into sampling noise and finite-sample effects; such dependence can be absorbed into $A(t)$ (e.g. $A(t) \propto 1/N$ in regimes where estimation dominates) without changing the co-design structure, since the only required property is the $1/m(t)$ scaling of the reducible term and the τ scaling of the irreducible floor.

Estimation and usable surrogates. The co-design procedure does not require full knowledge of $A(t)$ and $B(t)$ as matrices; it requires them only through the scalar contractions $\langle K(r, t; C), A(t) \rangle$ and $\langle K(r, t; C), B(t) \rangle$ that appear in the objective. Accordingly, we may estimate $A(t)$ and $B(t)$ (or directly these contractions) by any of the following standard routes: (i) *theoretical plug-in* in the Gaussian-linear model, where H_t and Σ_t are explicit functions of (C, σ_t, s_t) and the DSM parameterization; (ii) *empirical curvature/noise estimation* from training diagnostics, using moving averages of per-t gradient covariances together with (approximate) Fisher or Gauss–Newton curvature to form $H_t^{-1} \Sigma_t H_t^{-1}$; or (iii) *replicated runs*, where one measures sample covariances of the learned score parameters across independent trainings restricted to a fixed time bin, and regresses the observed variance against $1/m$ and τ to obtain $A(t)$ and $B(t)$. In practice we often further scalarize by assuming that, in the eigenbasis relevant to $K(r, t; C)$, both $A(t)$ and $B(t)$ are approximately diagonal or isotropic within spectral bands, which suffices for the bandpower reduction used later. Under any of these estimation procedures, the model (1) provides the missing link between training compute and the sampler-amplified score error term in $\mathcal{E}(r, \gamma(\cdot), m(\cdot))$.

5 The budgeted co-design problem

We now formalize the joint design of (i) the terminal reverse-time r (equivalently the forward-time truncation point $t = T - r$), (ii) the sampling step schedule $\gamma(\cdot)$, and (iii) the training allocation $m(\cdot)$, under a single compute budget. Throughout we view r as controlling the bias–variance tradeoff inherent to stopping sampling before reaching the smallest noise levels: smaller r reduces the truncation term $E^{(0)}(r)$ but typically increases both discretization burden and the kernel-amplified score-error accumulation.

Continuous-time co-design program. Let $c_{\text{train}}(t)$ denote the cost (in abstract compute units) per effective training update at noise level t, and let $c_{\text{samp}}(t, \gamma(t))$ denote the cost density of sampling at time t when using step size $\gamma(t)$ (this cost may be taken constant per step, or may depend on

γ if, e.g., adaptive correctors or higher-order methods are used). Given the kernelized surrogate

$$\mathcal{E}(r,\gamma(\cdot),m(\cdot)) = E^{(0)}(r)+E^{\mathrm{disc}}(r,\gamma(\cdot))+\int_r^T \gamma(t)\left\langle K(r,t;C), \frac{A(t)}{m(t)}+\tau B(t)\right\rangle dt,$$

our co-design problem is

$$\min_{r,\ \gamma(\cdot),\ m(\cdot)} \quad \mathcal{E}(r,\gamma(\cdot),m(\cdot)) \tag{2}$$

$$\text{s.t.} \quad \int_r^T c_{\mathrm{train}}(t)\, m(t)\, dt + \int_r^T c_{\mathrm{samp}}(t,\gamma(t))\, dt \ \le\ B, \tag{3}$$

$$m(t) \ge 0, \qquad \gamma(t) > 0, \qquad r \in [0,T], \tag{4}$$

$$\gamma(t) \text{ obeys solver feasibility/stability constraints for all } t \in [r,T]. \tag{5}$$

We emphasize that (2) is well-defined whenever $t \mapsto \langle K(r,t;C), A(t)\rangle$ is integrable on $[r,T]$ and $m(t) > 0$ on the set where this contraction is positive. In particular, if $m(t) = 0$ on a set of positive measure where $\langle K(r,t;C), A(t)\rangle > 0$ and $\gamma(t) > 0$, then the reducible term diverges; thus optimal solutions will either allocate $m(t) > 0$ where the kernel sensitivity is nonzero, or else rely on regimes where the contraction vanishes (e.g., directions/time regions rendered irrelevant by K).

Discretized-time version. In implementation we optimize over a finite grid. Fix a candidate stopping time r and a partition $r = t_1 < t_2 < \cdots < t_L = T$ with associated representative times $\{t_i\}_{i=1}^L$. We parameterize a piecewise-constant training allocation by nonnegative integers or reals $\{m_i\}_{i=1}^L$ (effective updates in bin i) and a sampling schedule by step sizes $\{\gamma_i\}_{i=1}^L$ (or, equivalently, by per-step values $\{\gamma_k\}_{k=1}^K$ when the grid is indexed by reverse-time steps). Writing $K_i(r) := K(r,t_i;C)$, $A_i := A(t_i)$, $B_i := B(t_i)$, and absorbing bin widths into γ_i if desired, a canonical Riemann approximation yields

$$\min_{r,\ \{\gamma_i\},\ \{m_i\}} \quad E^{(0)}(r) + E^{\mathrm{disc}}(r,\{\gamma_i\}) + \sum_{i=1}^L \gamma_i \left\langle K_i(r), \frac{A_i}{m_i} + \tau B_i\right\rangle, \tag{6}$$

subject to the discrete compute budget

$$\sum_{i=1}^L c_{\mathrm{train}}(t_i)\, m_i + \sum_{i=1}^L c_{\mathrm{samp}}(t_i,\gamma_i) \ \le\ B, \qquad m_i \ge 0, \quad \gamma_i > 0, \tag{7}$$

and solver feasibility constraints analogous to (5). This discretized form makes explicit that, conditional on r and $\{\gamma_i\}$, the training decision couples across times only through the single linear budget (7); conversely, conditional on $\{m_i\}$, the sampling decision trades off E^{disc} against the weighted contractions $\langle K_i(r), A_i/m_i + \tau B_i\rangle$.

Convexity structure and separability. The joint program (2)–(5) is not, in general, convex in the pair $(\gamma(\cdot), m(\cdot))$ once $E^{\mathrm{disc}}(r, \gamma(\cdot))$ and the sampling cost are included, nor is it convex in r due to the dependence of $K(r, t; C)$ and $E^{(0)}(r)$ on r. The crucial property we exploit is conditional convexity in $m(\cdot)$: for fixed $(r, \gamma(\cdot))$, the mapping

$$m(\cdot) \;\longmapsto\; \int_r^T \gamma(t) \left\langle K(r, t; C), A(t) \right\rangle m(t)^{-1} \, dt$$

is convex on $\{m(t) > 0\}$ because $x \mapsto 1/x$ is convex and $\gamma(t)\langle K(r, t; C), A(t)\rangle \geq 0$ whenever $K(r, t; C) \succeq 0$ and $A(t) \succeq 0$. The irreducible term $\int_r^T \gamma(t)\tau\langle K(r, t; C), B(t)\rangle dt$ does not depend on m and therefore does not affect convexity. This conditional convexity implies that, for each fixed $(r, \gamma(\cdot))$, the optimal training allocation admits KKT characterization and, under mild regularity, a closed-form rule. This will be the content of the next section.

Sampling feasibility and stability constraints. Finally, we specify the constraint (5). Since we restrict attention to explicit or semi-explicit solvers (Euler, Heun, low-order RK, or their ODE analogues), stability imposes upper bounds on γ. In the Gaussian-linear regime, the reverse-time dynamics are linear in x with a time-dependent drift matrix whose spectrum depends on $(\beta_t, \xi_t, \alpha, \sigma_t, s_t)$ and on the data spectrum $\{\lambda_i\}$. Consequently there exists a computable function $\gamma_{\max}(t; \lambda_{\max})$ such that requiring

$$0 < \gamma(t) \leq \gamma_{\max}(t; \lambda_{\max}) \tag{8}$$

ensures that each one-step update is well-defined and does not amplify errors catastrophically in the stiffest spectral direction. In practice (8) can be instantiated either by classical explicit stability criteria (e.g. $\gamma \lesssim 2/L(t)$ for an effective Lipschitz constant $L(t)$) or by solver-specific constraints derived from the linear test equation in each eigen-direction. These constraints interact with r: when the schedule becomes stiff near $t \downarrow 0$, stability may force $\gamma(t)$ to be prohibitively small, making early stopping (larger r) compute-optimal even when $E^{(0)}(r)$ is not negligible.

6 Optimal training allocation $m^*(t)$

We now fix a candidate stopping time r and a feasible sampling schedule $\gamma(\cdot)$, and solve for the compute-optimal training allocation $m(\cdot)$. Since the irreducible optimizer-noise term $\int_r^T \gamma(t)\tau\langle K(r, t; C), B(t)\rangle dt$ does not depend on $m(\cdot)$, the relevant subproblem is

$$\min_{m(\cdot)\geq 0} \int_r^T \frac{s(t)}{m(t)} \, dt \qquad \text{s.t.} \qquad \int_r^T c_{\mathrm{train}}(t) \, m(t) \, dt \leq B_{\mathrm{train}}, \tag{9}$$

where we abbreviate the (nonnegative) *kernel sensitivity*

$$s(t) \;:=\; \gamma(t)\,\big\langle K(r,t;C), A(t)\big\rangle \;\geq\; 0, \tag{10}$$

and where B_{train} is the training sub-budget remaining after accounting for sampling cost (either fixed a priori or induced by the outer optimization). We assume $s(\cdot)$ and $c_{\text{train}}(\cdot)$ are measurable and integrable on $[r,T]$, with $c_{\text{train}}(t) > 0$ almost everywhere. Note that if $s(t) > 0$ on a set of positive measure, then any feasible $m(\cdot)$ that vanishes on that set makes the objective in (9) diverge; thus optimal solutions necessarily satisfy $m(t) > 0$ almost everywhere on $\{t : \; s(t) > 0\}$.

KKT conditions and square-root rule. Problem (9) is convex because $m \mapsto s/m$ is convex on $(0,\infty)$ for each fixed $s \geq 0$, and the constraint set is linear. Writing the Lagrangian

$$\mathcal{L}(m,\lambda,\nu) \;=\; \int_r^T \frac{s(t)}{m(t)}\,dt + \lambda\Big(\int_r^T c_{\text{train}}(t)m(t)\,dt - B_{\text{train}}\Big) - \int_r^T \nu(t)\,m(t)\,dt,$$

with $\lambda \geq 0$ and $\nu(t) \geq 0$, the pointwise stationarity condition on $\{t : \; m(t) > 0\}$ yields

$$-\frac{s(t)}{m(t)^2} + \lambda c_{\text{train}}(t) - \nu(t) \;=\; 0. \tag{11}$$

Complementary slackness gives $\nu(t)m(t) = 0$ and $\lambda\big(\int c_{\text{train}}m - B_{\text{train}}\big) = 0$. On any t with $s(t) > 0$ we must have $m(t) > 0$ and hence $\nu(t) = 0$, so (11) implies

$$m^*(t) \;=\; \sqrt{\frac{s(t)}{\lambda\, c_{\text{train}}(t)}} \;=\; \sqrt{\frac{\gamma(t)\langle K(r,t;C), A(t)\rangle}{\lambda\, c_{\text{train}}(t)}} \qquad \text{for a.e. } t \in [r,T] \text{ with } s(t) > 0. \tag{12}$$

If $s(t) = 0$, then $m^*(t) = 0$ is optimal (any positive allocation would consume budget without reducing (9)). The multiplier λ is chosen so that the budget is saturated whenever $B_{\text{train}} > 0$ and $s \not\equiv 0$:

$$\int_r^T c_{\text{train}}(t)\,m^*(t)\,dt \;=\; B_{\text{train}}. \tag{13}$$

In particular, substituting (12) into (13) yields

$$\lambda \;=\; \left(\frac{\int_r^T \sqrt{s(t)c_{\text{train}}(t)}\,dt}{B_{\text{train}}}\right)^2, \qquad \text{and thus} \qquad m^*(t) \;=\; \frac{B_{\text{train}}}{\int_r^T \sqrt{s(u)c_{\text{train}}(u)}\,du}\sqrt{\frac{s(t)}{c_{\text{train}}(t)}}. \tag{14}$$

This is the continuous-time analogue of the familiar "water-filling" principle: budget concentrates where the geometric mean $\sqrt{s(t)c_{\text{train}}(t)}$ is large, and within those regions the allocation scales as a square root of sensitivity and inverse square root of cost.

Equal marginal gain interpretation. The rule (12) can be read as an equalization of marginal improvement per unit compute. Indeed, the reducible contribution at time t is $s(t)/m(t)$, so the infinitesimal decrease in objective from increasing $m(t)$ is

$$-\frac{\partial}{\partial m(t)}\Big(\frac{s(t)}{m(t)}\Big) \;=\; \frac{s(t)}{m(t)^2}.$$

Dividing by the marginal cost $c_{\mathrm{train}}(t)$, the KKT condition (11) (with $\nu(t) = 0$) is precisely

$$\frac{1}{c_{\mathrm{train}}(t)} \cdot \frac{s(t)}{m^*(t)^2} \;=\; \lambda, \tag{15}$$

i.e. the "error decrease per unit training compute" is constant across all times that receive positive allocation. Times with $s(t) = 0$ have zero marginal gain and are optimally ignored.

Discrete-time version and rounding. For the discretized objective (6)–(7), fixing $(r, \{\gamma_i\})$ yields

$$\min_{m_i \ge 0} \sum_{i=1}^{L} \frac{s_i}{m_i} \quad \text{s.t.} \quad \sum_{i=1}^{L} c_i m_i \le B_{\mathrm{train}}, \qquad s_i := \gamma_i \langle K_i(r), A_i\rangle,\;\; c_i := c_{\mathrm{train}}(t_i).$$

The KKT conditions give, for all i with $s_i > 0$,

$$m_i^* \;=\; \sqrt{\frac{s_i}{\lambda c_i}}, \qquad \text{with} \qquad \sum_{i=1}^{L} c_i m_i^* \;=\; B_{\mathrm{train}}. \tag{16}$$

If integer m_i are required, one may round m_i^* while preserving the total budget (e.g. via greedy adjustment by largest marginal gains s_i/m_i^2); convexity implies that such rounding incurs only a controlled additive increase in the surrogate.

Common cost models and clipped water-filling. When $c_{\mathrm{train}}(t) \equiv c_0$ is constant, (14) simplifies to

$$m^*(t) \;\propto\; \sqrt{s(t)} \;=\; \sqrt{\gamma(t)\langle K(r, t; C), A(t)\rangle},$$

so training concentrates at noise levels where the sampler is most sensitive (large γ and large kernel contraction) and where score estimation is intrinsically difficult (large $A(t)$ in the relevant directions). If additional box constraints are present, e.g. $0 \le m(t) \le m_{\max}(t)$ due to finite replay capacity or per-time data constraints, then the same KKT calculation yields a clipped rule

$$m^*(t) \;=\; \min\Big\{m_{\max}(t),\; \sqrt{\tfrac{s(t)}{\lambda c_{\mathrm{train}}(t)}}\Big\},$$

with λ adjusted so that the (possibly non-saturated) budget and active-set constraints are jointly satisfied; this is the precise sense in which the solution behaves like water-filling.

We shall henceforth regard $m^*(\cdot)$ as an explicit functional of $(r, \gamma(\cdot))$ via (12) (or (16)), and substitute it into the surrogate to reduce the co-design problem to the remaining choices of sampling schedule and stopping time.

7 Optimal sampling schedule $\gamma^*(t)$ and stopping time r^*

Having optimized $m(\cdot)$ for fixed $(r, \gamma(\cdot))$ in Section 6, we now turn to the remaining design variables: the sampling step-size schedule $\gamma(\cdot)$ and the terminal time r. We emphasize that $\gamma(\cdot)$ influences the surrogate both through the discretization bias $E^{\text{disc}}(r, \gamma)$ and through the kernel-weighted amplification of score error. In particular, for fixed r and fixed training budget B_{train}, substituting (14) into the reducible term yields the reduced functional

$$\mathcal{E}_{\text{red}}(r, \gamma) = E^{(0)}(r) + E^{\text{disc}}(r, \gamma) + \frac{1}{B_{\text{train}}} \left(\int_r^T \sqrt{\gamma(t) \, \langle K(r, t; C), A(t) \rangle \, c_{\text{train}}(t)} \, dt \right)^2 + \int_r^T \gamma(t) \, \tau \langle K(r, t; \tag{17}$$

where we have used $s(t) = \gamma(t) \langle K(r, t; C), A(t) \rangle$ and the optimal value of (9) equals $(\int \sqrt{s c_{\text{train}}})^2 / B_{\text{train}}$. While (17) is not pointwise separable in γ because of the squared integral, it makes explicit the basic monotonicity: decreasing $\gamma(t)$ reduces both score-error accumulation terms, whereas it typically increases sampling compute through a larger effective number of steps.

Sampling compute budget and KKT conditions. To state optimality conditions, we adopt a standard continuous relaxation in which sampling cost is proportional to step density:

$$\int_r^T \frac{c_{\text{samp}}(t)}{\gamma(t)} \, dt \ \le \ B_{\text{samp}}, \qquad 0 < \gamma(t) \le \gamma_{\max}(t) \ \text{ a.e. on } [r, T], \tag{18}$$

where $c_{\text{samp}}(t) > 0$ captures per-step cost (e.g. one model evaluation) and $\gamma_{\max}(t)$ encodes solver stability. For the discretization bias, we use the generic order-q proxy

$$E^{\text{disc}}(r, \gamma) \ \approx \ \int_r^T d_q(t) \, \gamma(t)^q \, dt, \tag{19}$$

with $q \ge 1$ the global order of the solver in the relevant weak/mean-square sense and $d_q(t) \ge 0$ a problem-dependent local smoothness coefficient (depending on the unified VP/VE drift and diffusion, and on the spectrum of C through stability constants). Fixing r and treating the training allocation as fixed (or as given by the previous section, in which case one may apply the conditions below within an outer iteration), the γ-subproblem takes the schematic form

$$\min_{\gamma} \int_r^T \Big(a(t)\gamma(t) + d_q(t)\gamma(t)^q\Big)\, dt \quad \text{s.t.} \quad \int_r^T \frac{c_{\text{samp}}(t)}{\gamma(t)}\, dt \le B_{\text{samp}},$$

where $a(t) := \langle K(r,t;C), A(t)/m(t) + \tau B(t)\rangle$ is the kernel-weighted score-error coefficient. The Lagrangian with multiplier $\eta \ge 0$ yields the pointwise stationarity condition on the active set $\{t : \ 0 < \gamma(t) < \gamma_{\max}(t)\}$:

$$a(t) \ + \ q\, d_q(t)\, \gamma(t)^{q-1} \ - \ \eta\, \frac{c_{\text{samp}}(t)}{\gamma(t)^2} \ = \ 0. \tag{20}$$

Together with complementary slackness for (18) and clipping at $\gamma_{\max}$, (20) characterizes $\gamma^*(t)$.

Two regimes are particularly transparent. If discretization dominates locally (so that $qd_q(t)\gamma(t)^{q-1} \gg a(t)$), then (20) gives the power law

$$\gamma^*(t) \ \asymp \ \Big(\tfrac{\eta\, c_{\text{samp}}(t)}{q\, d_q(t)}\Big)^{1/(q+1)}. \tag{21}$$

If instead kernel-weighted score error dominates (so that $a(t) \gg qd_q(t)\gamma(t)^{q-1}$), then

$$\gamma^*(t) \ \asymp \ \sqrt{\tfrac{\eta\, c_{\text{samp}}(t)}{a(t)}}. \tag{22}$$

In both cases η is set so that the sampling budget is saturated unless B_{samp} is so large that the optimum is clipped by $\gamma_{\max}$ everywhere.

Implications for VP/VE/EDM families. In the Gaussian-linear limit, the kernel $K(r,t;C)$ is explicit in the eigenbasis of C and typically amplifies score error more strongly at low noise. Consequently, across common schedules one may regard $a(t)$ as scaling like a negative power of the noise level (up to spectral weights), so (22) prescribes smaller step sizes precisely where σ_t is small. Concretely: (i) for VP schedules, one commonly has $\sigma_t^2 \asymp t$ near $t = 0$, leading to $a(t)$ that is approximately proportional to σ_t^{-2} (and in some parametrizations σ_t^{-4}); hence $\gamma^*(t)$ decays roughly like a positive power of σ_t as $t \downarrow 0$, producing a dense grid near the data end; (ii) for VE schedules with σ_t increasing polynomially (e.g. $\sigma_t \asymp t^\rho$ near 0), the same principle yields $\gamma^*(t) \asymp t^\rho$ (or $t^{2\rho}$) up to the cost weight $c_{\text{samp}}(t)$; (iii) for EDM-type schedules with $\sigma(t)$ a controlled power-law interpolation, the above behavior persists with exponents determined by the local power ρ in $\sigma_t \asymp t^\rho$ near 0, and clipping by $\gamma_{\max}$ captures the empirically observed need for smaller steps in stiff regions.

Solver order q and discretization decay. The proxy (19) implies that, at fixed step density, increasing q steepens the decay of E^{disc} with maximal step size. Under the budget model (18), the characteristic step size scales like $\bar{\gamma} \asymp (T-r)/K$ with K the effective number of steps, so E^{disc} behaves like $O(K^{-q})$ (modulo stiffness through d_q). Thus higher-order solvers convert sampling compute into accuracy more efficiently in smooth regimes, but the stability cap $\gamma_{\max}(t)$ and stiffness encoded in $d_q(t)$ can limit attainable gains, particularly near small σ_t where the reverse dynamics become ill-conditioned.

Stopping time r^* via bias–variance balance. Finally, r trades truncation bias against the accumulation of (optimized) score error and discretization effects. In the regime where E^{disc} is controlled (e.g. by allocating sufficient sampling budget), the leading balance is between $E^{(0)}(r)$ and the optimizer-limited term. Assuming near-zero scaling $\sigma_r^2 \asymp r^p$ and $E^{(0)}(r) \asymp \sigma_r^4 \asymp r^{2p}$, and that the dominant kernel-weighted accumulation behaves like $\tau_{\mathrm{eff}} \int_r^T \sigma_t^{-2}\,dt$, the minimizing r^* satisfies the scaling stated in Theorem 3:

$$r^* \asymp \tau_{\mathrm{eff}}^{1/(2+p)} \quad \text{up to logarithmic factors when } \int_r^T \sigma_t^{-2} dt \text{ diverges slowly (VP-like).}$$

For example, VP schedules yield a slow (logarithmic) divergence and hence a r^* that is a small power of τ_{eff} with an additional log correction; VE schedules with $\sigma_t^2 \asymp t^p$ and $p > 1$ yield a purely algebraic tradeoff; and EDM schedules inherit the exponent through the local power-law behavior of σ_t near 0.

Matching lower bounds: minimax perspective under compute constraints. We now justify that the surrogate error achieved by the optimized allocations is not merely an artifact of the upper-bound analysis, but is (in the Gaussian-linear model) unavoidable up to constants and mild logarithmic effects. Formally, we consider a class of algorithms that may (i) adaptively choose which noise levels to train on, (ii) use at most $m(t)$ effective stochastic gradient samples at each t (or time bin), and (iii) sample using any stable reverse-time solver stopped at terminal reverse-time r. The minimax question is: among all such procedures, how small can $\mathbb{E}[W_2^2(p_{\mathrm{data}}, q_r)]$ be as a function of $(r, \gamma(\cdot), m(\cdot))$ and the optimizer noise level τ?

Reduction of score learning at fixed t to Gaussian regression. Fix a noise level t. In the unified VP/VE formulation, the forward marginal has the affine form

$$x_t \;=\; s_t x_0 + \sigma_t z, \qquad x_0 \sim \mathcal{N}(\mu, C),\; z \sim \mathcal{N}(0, I),$$

so $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$ with $\mu_t = s_t\mu$ and $\Sigma_t = s_t^2 C + \sigma_t^2 I$. The exact score is linear:

$$s^*(x,t) \;=\; -\Sigma_t^{-1}(x - \mu_t).$$

Under our linear realizability assumption, learning the score at time t is equivalent to estimating the linear operator Σ_t^{-1} (and μ_t, which we subsume into an intercept parameter) from the training signal available at that t. In DSM, the population objective at time t is a quadratic form whose Hessian is the Fisher information of a Gaussian linear model; consequently, any estimator based on $m(t)$ i.i.d. samples (or $m(t)$ unbiased stochastic gradients with comparable noise) has a mean-squared error lower bounded by a constant multiple of $1/m(t)$ in the directions where the information is nondegenerate.

One may make this explicit by restricting to a one-dimensional subproblem in the eigenbasis of C. Writing $Cu_i = \lambda_i u_i$, the marginal variance along u_i equals $s_t^2\lambda_i + \sigma_t^2$. Along this coordinate, the score coefficient is $-(s_t^2\lambda_i + \sigma_t^2)^{-1}$. Estimating this scalar from $m(t)$ samples is a standard Gaussian mean/variance estimation problem; by Cramér–Rao (or Le Cam's two-point method on a local parametric subfamily), the variance of any unbiased (or sufficiently regular) estimator obeys

$$\operatorname{Var}\Big(\widehat{(s_t^2\lambda_i + \sigma_t^2)^{-1}}\Big) \;\geq\; \frac{c}{m(t)} \cdot \mathcal{I}_i(t)^{-1},$$

for an information factor $\mathcal{I}_i(t)$ depending only on $(s_t^2\lambda_i + \sigma_t^2)$. Aggregating across coordinates and lifting back to matrices yields a matrix MSE bound of the schematic form

$$\mathbb{E}\big[\|\widehat{\theta}_t - \theta_t^*\|_{H_t}^2\big] \;\geq\; \frac{c}{m(t)},$$

where $H_t \succeq 0$ is the DSM Hessian (a deterministic function of Σ_t) and $c > 0$ is universal in the Gaussian family. This is the origin of the $\Omega(1/m(t))$ term in Theorem 4.

Irreducible optimizer noise as a $\Omega(\tau)$ floor. If training is performed by a constant-step stochastic optimizer that reaches stationarity, then even with infinite data reuse at a fixed t, the stationary parameter fluctuations do not vanish. In the quadratic (Gaussian-linear) regime, SGD-like dynamics are well-approximated by a linear stochastic recursion, whose stationary covariance satisfies a discrete Lyapunov equation. Under standard regularity (e.g. step size below the stability threshold), this yields a lower bound of the form

$$\inf_{\text{constant-step stationary schemes}} \operatorname{Cov}(\theta_t - \theta_t^*) \;\succeq\; c'\,\tau\,B(t)$$

for an appropriate noise-shape matrix $B(t)$ determined by gradient noise, with $c' > 0$ depending only on stability constants. Thus, alongside the reducible $1/m(t)$ component, there is an irreducible $\Omega(\tau)$ component that no reweighting or additional compute at other times can remove.

Propagation of per-time estimation error to a W_2^2 lower bound. We next transfer these per-time limitations to a bound on the final sampling error. In the Gaussian-linear setting, the effect of an additive score error $e_t(x)$ on the terminal law q_r can be linearized, and $\mathbb{E}[W_2^2]$ admits a quadratic form in the score-error process. The sensitivity operator $K(r, t; C)$ (diagonalizable in the eigenbasis of C) maps the parameter covariance at time t into the contribution to the Bures/mean perturbations of the terminal Gaussian. Consequently, for any stable sampler with step schedule $\gamma(\cdot)$, we obtain a lower bound matching the structure of the upper surrogate:

$$\mathbb{E}[W_2^2(p_{\text{data}}, q_r)] \;\geq\; c_0 E^{(0)}(r) \;+\; c_1 \int_r^T \gamma(t) \left\langle K(r, t; C), \frac{A(t)}{m(t)} + \tau B(t) \right\rangle dt,$$

up to terms of the same order as the discretization bias (which cannot be made negative and is separately controlled by the sampling budget). Intuitively, $K(r, t; C)$ identifies the parameter directions that the sampler amplifies most strongly; the lower bound shows that errors in precisely these directions must persist given the compute constraints at time t.

Comparison to the achieved upper bound (constants and logs). Combining the lower bound above with the constructive upper bound from Theorem 1 shows that our co-designed schedules are minimax-optimal in order. In particular, under the linear training-cost model of Theorem 2, optimizing $m(\cdot)$ yields the upper value $(\int_r^T \sqrt{\gamma(t)\langle K(r, t; C), A(t)\rangle c_{\text{train}}(t)}\, dt)^2/B_{\text{train}}$, and the lower bound implies that no algorithm can improve the B_{train}^{-1} scaling or alter the kernel-weighted geometry encoded by $K(r, t; C)$, except by constant factors. The only systematic discrepancy arises in regimes where the kernel-weighted accumulation $\int_r^T \sigma_t^{-2} dt$ diverges slowly (VP-like schedules), in which case both upper and lower bounds inherit unavoidable logarithmic dependence on r (and, in discrete time, mild additional logs from grid regularity). Thus the remaining gap is limited to constants and such log factors, rather than any polynomial improvement, confirming that the co-design principle is sharp in the Gaussian-linear model.

Practical scheduler: estimable inputs and an implementable co-design routine. We now describe how the quantities appearing in the co-design objective can be estimated from finite data and training diagnostics, and how one may implement a scheduler that outputs a training weight $w(t)$ (equivalently $m(t)$ under fixed total updates), a sampling step-size schedule $\gamma(t)$ (or $\{\gamma_k\}_{k=0}^{K-1}$), and a stopping time r. The guiding principle is that all model-dependent information enters through (i) the power spectrum of C (or a bandpower approximation), (ii) the diffusion schedule (s_t, σ_t) and solver family (to compute the kernel K and discretization proxy), and (iii) per-noise-level training-noise statistics (to instantiate $A(t)$ and $B(t)$ or suitable scalar surrogates).

Estimating the spectrum or bandpower of C. Given samples $x^{(n)} \sim p_{\mathrm{data}}$, we form the empirical covariance $\widehat{C}$ (after centering). In high dimension, we typically avoid a full eigendecomposition and instead estimate bandpowers $\{(\Lambda_b, \rho_b)\}_{b=1}^{B}$, where Λ_b is a representative eigenvalue in band b and ρ_b is the multiplicity weight (fraction of energy or count). Concretely, one may obtain: (i) top-eigenpairs via randomized SVD/power iteration; (ii) the remaining bulk via stochastic trace estimation applied to resolvents, e.g. $\mathrm{tr}((\widehat{C}+\eta I)^{-1})$ across a grid of η, which can be inverted to fit a coarse spectral density; or (iii) direct binning of approximate eigenvalues when moderate d permits. For our purposes, a low-resolution bandpower is often sufficient because the kernel $K(r, t; C)$ varies smoothly in λ under typical schedules; we therefore recommend enforcing monotone bands and using conservative Lipschitz-based error bars (e.g. enlarge bands) when the estimate is noisy.

Computing kernel sensitivities from bandpowers. Once (s_t, σ_t) and the solver family are fixed, we precompute per-time-bin sensitivities

$$S_i(r) \; := \; \Big\langle K(r, t_i; C), A(t_i) \Big\rangle,$$

or, in a bandpower approximation, replace the eigen-sums by band-sums of the form

$$S_i(r) \; \approx \; \sum_{b,b'} \rho_b \rho_{b'} \, k_{r,t_i}(\Lambda_b, \Lambda_{b'}) \, a_{b,b'}(t_i),$$

where $k_{r,t}$ denotes the spectral kernel induced by the sampler and $a_{b,b'}(t)$ is the representation of $A(t)$ in the same spectral basis (often diagonal or nearly so in Gaussian-linear surrogates). In practice, we frequently scalarize further and use $S_i(r) \approx \mathrm{tr}(K(r, t_i; C)) \, \bar{a}(t_i)$, where $\bar{a}(t)$ is a one-number proxy for training difficulty at noise level t (see below). This reduces preprocessing to $O(LB)$ for L time bins.

Estimating training-noise statistics $A(t)$ and $B(t)$ (or their surrogates). The scheduler only requires the kernel-weighted contractions $\langle K, A\rangle$ and $\langle K, B\rangle$. We therefore advocate estimating scalar proxies

$$a(t) \; \approx \; \langle K(r, t; C), A(t)\rangle, \qquad b(t) \; \approx \; \langle K(r, t; C), B(t)\rangle,$$

directly from training logs, rather than attempting to recover full matrices. Two practical options are:

- *Gradient-noise route:* during training, at each t-bin we periodically collect mini-batch gradients g of the DSM loss, estimate $\widehat{\mathrm{Cov}}(g)$, and approximate the stationary parameter covariance via a quadratic-model Lyapunov proxy. This yields an empirical decomposition into a reducible component scaling as $1/m(t)$ and an irreducible component scaling with τ.

- *Learning-curve route:* run short pilot trainings at fixed t (or a small set of t-bins), measure score-error proxies (e.g. validation DSM loss, or denoising MSE) as a function of the number of updates, and fit a two-term model $\mathrm{err}(m) \approx \kappa_1(t)/m + \kappa_2(t)\tau$. The fitted κ_1, κ_2 serve as $a(t), b(t)$ up to multiplicative constants absorbed by λ in the KKT allocation.

Both approaches are robust to moderate model mismatch because the scheduler depends primarily on relative magnitudes across t, not absolute calibration.

Closed-form training weights and discretization. Fix r and a provisional sampling schedule $\gamma(t)$. Discretize $[r, T]$ into bins $\{t_i\}_{i=1}^{L}$ and let m_i denote the number of effective updates allocated to bin i, with cost $\sum_i c_{\mathrm{train}}(t_i) m_i \le B_{\mathrm{train}}$. The KKT solution gives

$$m_i^* \;=\; \sqrt{\frac{\gamma(t_i)\, a(t_i)}{\lambda\, c_{\mathrm{train}}(t_i)}}, \qquad w_i^* \;:=\; \frac{m_i^*}{\sum_j m_j^*} \;\propto\; \sqrt{\frac{\gamma(t_i)\, a(t_i)}{c_{\mathrm{train}}(t_i)}}.$$

We then implement training by sampling noise levels from the discrete distribution w_i^* (or by reweighting the loss accordingly). To avoid degenerate behavior under estimation noise, we impose floors and caps $w_{\min} \le w_i \le w_{\max}$, followed by renormalization, which is equivalent to restricting m_i to a compact feasible set.

Sampling schedule and stopping time selection. Given a training allocation (or under an alternating scheme), we choose $\gamma(t)$ and r by evaluating the surrogate

$$\widehat{\mathcal{E}}(r, \gamma, w) \;=\; E^{(0)}(r) + E^{\mathrm{disc}}(r, \gamma) + \sum_i \gamma(t_i)\Big(\frac{a(t_i)}{m_i} + \tau\, b(t_i)\Big),$$

under the sampling-cost constraint and solver stability bounds (which yield per-bin upper limits on $\gamma(t_i)$). Practically, we search over a small grid of r-candidates, and for each r we choose K and a monotone step schedule (e.g. uniform in a transformed time variable) that approximately minimizes E^{disc} per sampling compute, then re-solve for w using the KKT rule. This yields an implementable outer loop with a few iterations; empirically, one or two alternations suffice because the dependence of w on γ is smooth through $\sqrt{\gamma(\cdot)}$.

Robustness considerations beyond the Gaussian-linear surrogate. When the data are not exactly Gaussian and the score model is not exactly linear, the kernelized objective should be regarded as a structured heuristic. Nonetheless, several robustness devices preserve its qualitative behavior: (i) use bandpowers rather than full spectra to prevent overfitting to spurious eigenvalues; (ii) smooth $a(t)$ and $b(t)$ across adjacent t-bins (e.g. total-variation or spline smoothing) to reduce variance; (iii) enforce minimum training mass on intermediate noise levels to prevent catastrophic forgetting; and (iv) validate the chosen r by monitoring an observable proxy (e.g. sample quality vs. truncated reverse-time) and adjusting within the predicted basin. Under these safeguards, the scheduler typically produces stable curricula concentrating training where the sampler is most sensitive, while allocating sufficient sampling resolution where discretization error dominates.

Experiments (recommended): validating tightness, isolating gains, and testing predictivity. We recommend an experimental suite whose purpose is not merely to improve sample quality, but to validate the structural claims encoded by the surrogate objective $\mathcal{E}(r, \gamma, m)$: (i) kernelized propagation of score error to terminal W_2^2; (ii) the $1/m(t)$ reducible scaling and τ-limited irreducible floor; (iii) the KKT "equal-marginal-gain" allocation rule; and (iv) the predicted compute–quality frontier as B varies. We organize the experiments by increasing model mismatch: from exactly Gaussian-linear settings (where all terms are measurable) to medium-scale diffusion/EDM models (where $\mathcal{E}$ is a structured predictor).

(A) Synthetic anisotropic Gaussians: end-to-end tightness under controlled spectra. We begin with $p_{\text{data}} = \mathcal{N}(0, C)$ in dimension $d \in \{64, 256, 1024\}$, with spectra chosen to stress anisotropy: (1) power-law $\lambda_i \propto i^{-\nu}$ for $\nu \in [0, 2]$; (2) spiked models with a few large eigenvalues and a flat bulk; and (3) banded spectra designed so that different eigen-bands dominate at different noise levels through the kernel $K(r, t; C)$. For each C, we fix a diffusion schedule (s_t, σ_t) (VP and VE are both instructive) and a solver family (Euler and Heun suffice to expose discretization effects). In this setting, the reverse-time SDE with an inexact linear score induces a terminal Gaussian $q_r = \mathcal{N}(\widehat{\mu}_r, \widehat{C}_r)$, and we can compute the *true* discrepancy $W_2^2(p_{\text{data}}, q_r)$ exactly via the Gaussian Bures formula. We then compare it to $\mathcal{E}(r, \gamma, m)$ with Rem treated as an empirical residual.

(A1) Measuring the training-noise model. To isolate the $\frac{A(t)}{m(t)} + \tau B(t)$ structure, we train a realizable linear score at each time bin t_i (either independently, or by freezing all but a time-embedding head) with constant-step SGD. We then estimate $\text{Cov}(\theta_{t_i} - \theta^*_{t_i})$ directly from multiple independent runs and verify the scaling

$$\operatorname{tr}\operatorname{Cov}(\theta_{t_i} - \theta^*_{t_i}) \approx \frac{c_1(t_i)}{m_i} + c_2(t_i)\tau,$$

as well as the kernel-weighted variant $\langle K(r, t_i; C), \mathrm{Cov}(\theta_{t_i} - \theta^*_{t_i})\rangle$ used by the scheduler. We recommend sweeping τ over at least one decade and m_i over a range where stationarity is achieved (verified by flat training-loss traces).

(A2) Tightness of kernel propagation and equal-marginal-gain. With $a(t_i) \approx \langle K(r, t_i; C), A(t_i)\rangle$ and $b(t_i) \approx \langle K(r, t_i; C), B(t_i)\rangle$ measured as above, we run the co-design routine to produce $w^*_i \propto \sqrt{\gamma(t_i)a(t_i)/c_{\mathrm{train}}(t_i)}$ and a sampling schedule γ. We then validate: (i) $\widehat{\mathcal{E}}$ predicts the ordering of candidate (r, γ, w) configurations; (ii) the realized allocation satisfies the KKT identity $m^{*2}_i c_{\mathrm{train}}(t_i)/(\gamma(t_i)a(t_i)) \approx$ const on active bins; and (iii) improvements concentrate precisely in regimes where $\langle K, A\rangle$ is large (e.g. near small σ_t for VP). For interpretability, we recommend plotting bandwise contributions $\sum_{b,b'} \rho_b \rho_{b'} k_{r,t}(\Lambda_b, \Lambda_{b'})$ to show which spectral components drive the allocation.

(B) Ablations: disentangling training reweighting from sampling improvements. To separate causes, we recommend three matched-compute conditions: (i) *training-only*: optimize $w(t)$ via KKT with a fixed sampler (fixed r and γ); (ii) *sampling-only*: optimize γ and r under a fixed training distribution (uniform w or a standard heuristic); (iii) *joint*: alternate until convergence (typically one or two passes). In each case, keep total compute $B = B_{\mathrm{train}} + B_{\mathrm{samp}}$ fixed and report both the terminal error and the decomposition of $\widehat{\mathcal{E}}$ into $E^{(0)}$, E^{disc}, and the score-error term. This ablation directly tests whether gains come from reallocating updates toward sensitive noise levels (large $\gamma\langle K, A\rangle$) versus from resolving discretization bottlenecks (large E^{disc}).

(C) Medium-scale diffusion/EDM: predictive power under model mismatch. We next test whether the scheduler remains predictive when the Gaussian-linear assumptions fail. We recommend CIFAR-10 and one higher-resolution dataset (e.g. FFHQ 64^2 or ImageNet 64^2) with an EDM-style preconditioning or a VP baseline. Since C in pixel space is both large and not the correct representation, we use *bandpower proxies* computed in a fixed feature space $\phi(x)$ (e.g. low-frequency DCT blocks, a frozen encoder, or early-layer activations of the score network) and treat the spectrum of $\mathrm{Cov}(\phi(x))$ as a stand-in for the relevant anisotropy. We then estimate $a(t)$ and $b(t)$ via a learning-curve route on a small pilot budget: for each t-bin, run short segments of training, fit $\mathrm{err}(m) \approx \kappa_1(t)/m + \kappa_2(t)\tau$, and use κ_1, κ_2 in place of $\langle K, A\rangle, \langle K, B\rangle$. The primary evaluation is whether $\widehat{\mathcal{E}}$ correlates with downstream metrics (FID, precision/recall, or likelihood proxies) across schedules at fixed compute, and whether the learned weights $w(t)$ exhibit stable qualitative shifts (e.g. increased mass at low noise when sampling is sensitive there).

(D) Compute–quality frontier: predicted shifts with B, τ, and solver order. Finally, we recommend a controlled sweep over total compute B and optimizer step size τ, reporting the Pareto frontier of quality versus compute for (i) a baseline schedule (uniform w, standard sampler), (ii) training-only optimized, (iii) sampling-only optimized, and (iv) joint co-design. The main check is that the frontier shifts are consistent with the surrogate scaling: diminishing returns in B_{train} due to the irreducible $\tau B(t)$ term, and movement of the optimal stopping time r^* as τ increases (consistent with the balancing logic underlying Theorem 3). We recommend also varying solver order q (Euler versus Heun) to confirm that when E^{disc} dominates, the scheduler reallocates budget toward sampling resolution rather than toward additional training at insensitive noise levels.

Limitations and extensions: correlated-time errors, non-Gaussianity, and connections to distillation/solvers. Our surrogate objective $\mathcal{E}(r, \gamma, m)$ is derived under a *time-decoupled* training-noise model in which each noise level t admits its own parameters θ_t (or, equivalently, the covariance of the score error is treated as block-diagonal across time bins). This abstraction is faithful in the Gaussian-linear toy setting and in diagnostic regimes where one can train per-t heads, but it is not literally satisfied by standard diffusion models that share a single network across all t. In the shared-parameter setting, the random vector $\theta - \theta^*$ induces *correlated* score errors $e_t(x)$ across t, and the terminal error depends on their joint covariance rather than on $\mathrm{Cov}(e_t)$ alone. A minimal extension replaces the single-time integrand by a quadratic form over times,

$$\int_r^T \int_r^T \Gamma(t, t'; r, \gamma) \left\langle \widetilde{K}(r; t, t'; C),\ \mathrm{Cov}(\theta - \theta^*) \right\rangle dt\, dt',$$

where Γ encodes the discretization weights (reducing to $\gamma(t)\delta_{t=t'}$ in the decoupled approximation) and $\widetilde{K}$ is the corresponding two-time sensitivity operator obtained by composing the sampler's linear response maps at times t and t'. Even in Gaussian-linear regimes this operator is explicit in the eigenbasis of C, but the *optimization* changes qualitatively: allocating compute at time t reduces error at *all* times through shared parameters, so the one-dimensional KKT rule of Theorem 2 is no longer exact.

A principled way to capture these correlations is to move from a per-time covariance ansatz to a *global* linearization (NTK-style) in which the score network is linear in θ around θ^*, and each t-bin contributes a Fisher/feature matrix $F(t)$ so that the (reducible) stationary covariance takes the schematic form

$$\mathrm{Cov}(\theta - \theta^*) \approx \Big(\int_r^T m(t)\, F(t)\, dt \Big)^{-1} \quad \text{(up to optimizer-noise terms).}$$

Substituting such a model yields an objective resembling optimal experimental design:

$$\min_{m(\cdot)\geq 0} \left\langle H(r,\gamma),\ \left(\int_r^T m(t)F(t)\,dt\right)^{-1}\right\rangle \quad \text{s.t.} \quad \int_r^T c_{\text{train}}(t)m(t)\,dt \leq B_{\text{train}},$$

for an appropriate PSD matrix $H(r,\gamma)$ summarizing sampler sensitivities. Unlike the separable $\int a(t)/m(t)\,dt$ structure, this objective is not generally reducible to a pointwise formula for $m^*(t)$; however, it remains convex in many linear models because $X \mapsto \langle H, X^{-1}\rangle$ is convex on $X \succ 0$, and $X = \int m(t)F(t)\,dt$ is affine in m. Thus, while the closed form may be lost, the *co-design viewpoint* persists: the relevant object is the interaction between a sampler-induced sensitivity H and a training-induced information accumulation $\int mF$. Practically, one may approximate this coupled problem by (i) coarse time binning with a low-rank model of cross-time covariances, (ii) block-diagonal surrogates calibrated by measuring correlations of gradient features across t, or (iii) alternating minimization over m and a parametric approximation of $\mathrm{Cov}(\theta - \theta^*)$.

A second limitation is the reliance on global Gaussianity to obtain an explicit kernel $K(r,t;C)$ and an exact W_2^2 evaluation via the Bures formula. For non-Gaussian p_{data}, the sampler output is not Gaussian even with an exact score, and score errors propagate through nonlinear dynamics. Nevertheless, the *local* linearization principle suggests a workable extension: along the forward noising path, each marginal p_t is often closer to Gaussian than p_{data}, and the reverse dynamics can be linearized around typical trajectories. In this regime, one may replace the single covariance C by a time-dependent local covariance $C_t := \mathrm{Cov}(x_t)$ (or its bandpower proxy in a feature space $\phi(x_t)$), and treat $K(r,t;C)$ as an empirical sensitivity operator $K_{\text{loc}}(r,t)$ estimated from Jacobian-vector products of the sampler with respect to score perturbations. The resulting surrogate retains the same formal structure,

$$\mathcal{E}_{\text{loc}}(r,\gamma,m) \approx E^{(0)}(r) + E^{\text{disc}}(r,\gamma) + \int_r^T \gamma(t)\left\langle K_{\text{loc}}(r,t),\, \tfrac{A_{\text{loc}}(t)}{m(t)} + \tau B_{\text{loc}}(t)\right\rangle dt,$$

but now $K_{\text{loc}}, A_{\text{loc}}, B_{\text{loc}}$ are *estimated* rather than derived. The principal caveat is that higher-order terms (ignored in Rem) need not be small at low noise, so predictivity depends on whether the sampler's linear response dominates the nonlinear coupling.

Finally, it is useful to relate co-design to two practice-driven choices: distillation and solver selection. In distillation (e.g. progressive distillation, consistency training), one trades a costly teacher sampler for a cheaper student while attempting to preserve terminal distribution. Our framework suggests that the teacher's compute should be spent where the student is most sensitive: the kernel $\gamma(t)K(r,t;\cdot)$ induces a principled reweighting over t for teacher training (and for student regression targets) analogous to the KKT allocation rule, but now interpreted as *importance weighting* for matching the teacher's denoising field. Conversely, solver choice (Euler versus higher-order methods, ODE solvers, adaptive step control) enters only through $E^{\mathrm{disc}}(r,\gamma)$, the stability region for γ, and the induced $\gamma(t)$ weighting of score error. This makes explicit a qualitative tradeoff often observed empirically: when discretization dominates, it is compute-optimal to raise solver order or allocate more sampling steps; when score error dominates, it is compute-optimal to allocate training updates to time regions with large kernel sensitivity and low training cost. Extending the co-design problem to include discrete solver-family choices (order q, adaptive controllers) leads to a mixed discrete–continuous optimization, but the surrogate provides a natural selection criterion by comparing the marginal return of reducing E^{disc} against the marginal return of reducing the kernel-weighted score error under the same compute.

Conclusion: kernelized compute-optimal co-design as a unifying perspective; open problems. We have advocated a co-design viewpoint in which *training* and *sampling* schedules are chosen jointly under an explicit compute budget, rather than tuned independently. In the Gaussian-linear regime, this viewpoint admits a concrete expression: the terminal discrepancy between the data distribution and the sampler output can be decomposed into a truncation term, a discretization term, and a kernel-weighted propagation of score-estimation error,

$$\mathcal{E}(r,\gamma,m) = E^{(0)}(r) + E^{\mathrm{disc}}(r,\gamma) + \int_r^T \gamma(t) \left\langle K(r,t;C),\ \tfrac{A(t)}{m(t)} + \tau B(t) \right\rangle dt,$$

with $K(r,t;C)$ capturing the sampler-induced sensitivity of the terminal law to score perturbations injected at time t. This kernel is the unifying object: it converts modeling and optimization noise into a *task-relevant* scalar cost, and thereby endows training updates at different noise levels with comparable units. In particular, the optimal allocation in the decoupled-time model follows an equal-marginal-gain principle (Theorem 2), with the closed-form rule

$$m^*(t) \ \propto\ \sqrt{\frac{\gamma(t)\, \langle K(r,t;C), A(t)\rangle}{c_{\mathrm{train}}(t)}},$$

showing that compute concentrates where (i) the sampler is sensitive, (ii) reducible training variance is large in sensitive directions, and (iii) per-update cost is favorable. The same surrogate also makes explicit why stopping early at reverse-time $r > 0$ can be compute-optimal: one trades truncation bias $E^{(0)}(r)$ against the accumulated effect of (partly irreducible) score error amplified by the kernel, leading to scaling laws for r^* (Theorem 3) and matching-order lower bounds (Theorem 4) within the model class.

Beyond providing closed-form schedules in a toy setting, we view the kernelized surrogate as a *conceptual reduction.* First, it clarifies what information about the data distribution is actually needed for schedule design: not the full density p_{data}, but (in this regime) a spectral proxy for C and a mechanism for estimating the training-noise terms $A(t), B(t)$. Second, it separates algorithmic choices into orthogonal levers: solver family and step sizes primarily shape E^{disc} and the weighting $\gamma(t)$, while training allocation controls the reducible part $A(t)/m(t)$ up to the optimizer-imposed floor $\tau B(t)$. Third, it places a variety of empirical heuristics—noise-level reweighting, curriculum learning, and hand-designed step schedules—under a single variational objective, making it possible to compare them on a common scale: marginal error decrease per unit compute.

Several open problems remain before this perspective becomes a predictive design tool for modern, shared-parameter diffusion models. *(i) Kernel estimation and robustness.* Even if one accepts a linear-response surrogate, estimating $K(r, t; \cdot)$ (or its low-rank summaries) accurately and cheaply for non-Gaussian data is nontrivial. A practical theory should quantify how errors in K affect the resulting schedules, and should provide stable estimators based on Jacobian–vector products, randomized probes, or bandpower approximations, with guarantees in regimes where higher-order remainder terms are controlled. *(ii) Shared parameters and cross-time coupling.* As discussed in the preceding section, time-correlated score errors invalidate the pointwise $1/m(t)$ structure. While convexity often survives under global linearizations, one needs scalable approximations (e.g. low-rank feature models, block-diagonal relaxations, or Nyström methods) and a principled understanding of when the decoupled KKT rule is a good approximation. *(iii) Nonstationary and adaptive training dynamics.* The stationary covariance model captures a compute-limited regime but ignores transient effects, momentum, learning-rate schedules, and nonconstant batch noise. A more faithful theory would treat τ and $m(t)$ as time-varying controls and would account for finite-horizon optimization error, ideally yielding a dynamic-programming or continuous-time control characterization rather than a static allocation. *(iv) Discrete solver choices and stability-limited regimes.* Our formulation accommodates solver order through E^{disc} and stability constraints, but a complete co-design must treat solver selection (Euler/Heun/RK/ODE, adaptive controllers, error estimators) as discrete decisions coupled to the training plan. This yields mixed discrete–continuous optimization, for which one would like approximation guarantees and interpretable selection criteria based on marginal returns. *(v) Objectives beyond W_2^2.* The Gaussian-linear analysis privileges W_2^2 because it is explicit and stable under small perturbations. In applications, one often optimizes perceptual or downstream-task metrics. Establishing when such metrics admit kernelized surrogates (or when they are Lipschitz with respect to W_2 over relevant model classes) would clarify the scope of schedule optimality claims. *(vi) Minimax limits* *under structural assumptions.* The matching lower bounds we obtain rely on Gaussian-linear structure. Extending lower bounds to non-Gaussian but structured families (e.g. log-concave measures, mixtures with spectral decay, manifold-supported data with intrinsic dimension) would sharpen the distinction between schedule-induced gains and information-theoretic barriers.

We thus regard kernelized compute-optimal co-design as a framework that (a) makes the tradeoffs between training, sampling, and stopping explicit; (b) yields exact optimality statements in a tractable regime; and (c) suggests a set of measurable primitives—sensitivities, training-noise covariances, and costs—that can be estimated and optimized even when the full generative model is far from linear. The primary remaining challenge is to develop estimators and robustness theory for these primitives in realistic settings, so that co-design moves from an explanatory surrogate to a reliable engineering principle.