SUN: STOCHASTIC UNBIASED MOMENTUM VIA FILTER DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Momentum is a central component of deep learning optimizers, yet its gradient-tracking behavior in practical training remains incompletely understood. Naturally, averaging past stochastic gradients suppresses noise, but the temporal center of mass of this weighted average lies in past iterations, causing the estimate to lag behind the current iteration. We interpret this source of gradient-tracking bias through the *group delay* of a discrete-time filter and analyze the tradeoff between delay and noise attenuation using the filter's frequency response.

Building on this interpretation, we propose a filter-design approach that uses amplitude response and group delay to guide the construction and tuning of momentum estimators. We instantiate this approach with SUN, a family of momentum filters that apply filtered gradient lookahead to reduce group delay in selected frequency bands while retaining noise attenuation. This design enables targeted phase compensation beyond the uniform lookahead used in existing methods. Experiments on language model pre-training indicate that the resulting momentum estimators improve training performance, supporting filter design as a practical approach to momentum-based optimization.

1 INTRODUCTION

Momentum is a fundamental component of optimization algorithms that incorporates past gradient information into the current update. Classical examples include Polyak's heavy-ball method (Polyak, 1964) and Nesterov's accelerated gradient method (Nesterov, 1983), while modern optimizers such as Adam and Muon (Kingma & Ba, 2014; Jordan et al., 2024) have made momentum a standard ingredient of deep learning. In stochastic optimization, averaging gradients across iterations can suppress sampling noise and stabilize the update direction. However, these gradients are evaluated at past iterates, so stronger averaging can also increase the lag in tracking the current gradient. Despite extensive theoretical study and widespread empirical success, how this tradeoff governs momentum's behavior in practical deep learning remains incompletely understood.

Existing optimization-based analyses provide useful theoretical guarantees for momentum, but their scope is often either too broad to explain its practical behavior or too narrow to guide its use in deep learning. Prior work established worst-case convergence bounds over general classes of smooth objectives, including strongly convex and nonconvex problems (Liu et al., 2020). These worst-case analyses show that stochastic gradient descent with momentum can match the convergence rates of SGD, but such broad guarantees do not explain the empirical benefits of momentum or how different averaging rules affect gradient tracking during training. Another line of work obtains sharper characterizations in structured settings, such as least-squares regression, identifying conditions under which stochastic acceleration is possible and limitations of classical momentum schemes (Jain et al., 2018; Kidambi et al., 2018). However, these conclusions depend on the particular objective structure and stochastic model, so their implications for the changing gradient dynamics of deep neural networks are less direct. Thus, both perspectives leave limited practical guidance for choosing how to combine past gradients to balance noise suppression and tracking delay, and practitioners are stuck with golden default values (e.g., $\beta = 0.95$ for Muon (Liu et al., 2025a)) when deploying optimization algorithms for a wide range of problems.

Meanwhile, recent work on optimizers for language model pre-training has emphasized the construction and scaling of update directions, often retaining EMA-based momentum as a fixed component

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(Jordan et al., 2024; Liu et al., 2025a). For example, Muon emphasizes matrix orthogonalization and update scaling (Liu et al., 2025a), SOAP applies Adam in the eigenbasis of Shampoo's preconditioner (Vyas et al., 2024), and Sophia combines diagonal Hessian preconditioning with update clipping (Liu et al., 2024). These methods develop distinct mechanisms for constructing update directions while retaining EMA-based gradient averaging as their momentum mechanism. Indeed, recent large-scale benchmarks compare different ways of constructing update directions across model scales and training settings (Wen et al., 2025; Zhao et al., 2024). These studies tune optimizer hyperparameters, including momentum coefficients, while retaining prescribed EMA-based gradient averaging. However, general principles for designing how momentum combines past gradients to balance noise suppression and tracking delay remain insufficiently understood. A few recent works, including AdEMAMix and MARS, have explored modifications to momentum by changing how past gradients are combined (Pagliardini et al., 2025; Yuan et al., 2024). However, these modifications are developed on a method-by-method basis, and a systematic framework for relating their design choices to important metrics such as gradient tracking and noise suppression remains limited.

Our Contributions. To balance noise suppression and gradient-tracking delay, we propose a *filter-design approach* for modeling, understanding, and improving momentum. We view the sequence of stochastic gradients as the input signal and the momentum estimate as the filter output. In this formulation, standard EMA can be viewed as a *low-pass filter*, where its amplitude response characterizes how gradient fluctuations are attenuated, while its group delay characterizes the lag introduced across frequencies. These quantities provide interpretable measures for comparing momentum estimators and expose the tradeoff between noise suppression and gradient tracking. Our approach uses these measures to guide the construction and tuning of momentum filters, targeting reduced delay in selected frequency bands while retaining noise attenuation.

More specifically, conventional smoothness-based analyses bound the magnitude of gradient changes by the distance between successive iterates and use these bounds to control momentum's tracking error (Liu et al., 2020). Such bounds do not describe how the delay introduced by momentum varies across frequencies. Our analysis explicitly characterizes this frequency dependence through *group delay*, which measures the lag introduced by the momentum filter at each frequency in units of iterations. Formally, group delay is the negative slope of the filter's phase response and describes the approximate shift of slowly varying signal envelopes near a given frequency. For EMA, this reveals that slowly varying, low-frequency gradient components experience greater delay than rapidly varying, high-frequency components. Together with the amplitude response, this characterization leads to an important filter (momentum) design principle: we shall seek to reduce delay in frequency bands containing useful gradient variations while retaining attenuation in bands dominated by noise.

We further utilize this framework to compare the temporal filtering mechanisms underlying AdEMAMix, Muon, and MARS (Pagliardini et al., 2025; Jordan et al., 2024; Yuan et al., 2024; Liu et al., 2025b). This comparison makes explicit how different uses of gradient history change the tradeoff between noise attenuation and tracking delay. For example, Figure 1 illustrates how increasing MARS's lookahead coefficient reduces delay at the cost of weaker noise attenuation, a tradeoff formalized in Section 4.1.

We instantiate this filter-design approach with SUN (Stochastic Unbiased momeNtum), a family of momentum estimators that use filtered gradient lookahead to shape the tradeoff between delay and noise attenuation. Filtering the lookahead signal lets us control where phase compensation is applied, reducing delay in selected frequency bands while limiting the amplification of high-frequency noise. Here, "unbiased" expresses our goal of compensating for the temporal bias described above by re-centering the estimate toward the current iteration. Our language model pre-training experiments indicate that the resulting momentum estimators improve training performance, providing empirical support for this filter-design approach.

Our main contributions are as follows:

- **Unified characterization of momentum.** We establish a common frequency-domain analysis of EMA, AdEMAMix, Muon, and MARS, connecting different ways of using gradient history with noise attenuation and tracking delay.
- **Constructive filter design.** We translate this analysis into a momentum design approach and instantiate it with SUN. By making gradient lookahead a designable filter, SUN enables targeted delay compensation while controlling noise amplification.

- **Validation in language model pre-training.** Our experiments demonstrate improved training performance with the resulting momentum estimators, validating the practical benefits of the proposed filter-design approach in the evaluated settings.

2 Related Works

Filtering and control views of optimization. Control-theoretic approaches model optimization algorithms as feedback systems and use stability conditions to analyze convergence and design algorithmic recursions (Lessard et al., 2016). Within this perspective, Mohammadi et al. (2019) quantify the tradeoff between accelerated convergence and noise amplification for smooth strongly convex problems. At the gradient-estimation level, Chen et al. (2020) use a dynamics model and Kalman filtering to construct curvature-corrected, noise-adaptive gradient estimates. Our focus is the temporal filter mapping stochastic gradients to momentum, with amplitude response and group delay serving as design criteria for balancing noise attenuation and tracking lag.

Frequency-domain momentum design. Closely related work by Li et al. (2025) interprets momentum as a time-varying gradient filter, analyzes its frequency response under a piecewise-constant approximation, and develops FSGDM by scheduling its magnitude response. Our approach emphasizes group delay and the design of filtered lookahead to compensate for delay in selected frequency bands while retaining noise attenuation. For matrix-based optimization, Li et al. (2026) show that momentum can stabilize Muon's singular subspaces under a structured signal-plus-perturbation model. Their analysis concerns denoising before matrix orthogonalization, whereas we study the temporal amplitude and delay responses of momentum recursions.

Variance-reduced gradient estimators. SVRG uses a reference gradient as a control variate to reduce sampling noise (Johnson & Zhang, 2013). SARAH and SPIDER recursively track gradients using differences evaluated at successive iterates on the same sampled data, together with periodic estimator refreshes (Nguyen et al., 2017; Fang et al., 2018). STORM combines momentum with such gradient-difference corrections to achieve variance reduction without periodic large-batch refreshes (Cutkosky & Orabona, 2019). MARS adapts scaled recursive momentum to preconditioned deep-learning optimizers; its practical approximation reuses the preceding stochastic gradient to avoid an additional gradient evaluation (Yuan et al., 2024). This change matters analytically, as differences of gradients evaluated on the same sample can partially cancel sampling noise, while differences between gradients from successive minibatches also contain changes in that noise. Our filter-design approach examines how these temporal corrections affect both noise attenuation and tracking delay, and generalizes the lookahead correction by filtering it before it enters the momentum average.

3 Frequency Analysis of Momentum Filters

Notation. We will assume $\mathbf{m}^t$ denotes a gradient estimator at iteration t that is used by a generic optimization algorithm $\mathbf{x}^{t+1} = \mathcal{A}_t(\mathbf{x}^t, \mathbf{m}^t)$, e.g., the gradient descent algorithm such that $\mathcal{A}_t(\mathbf{x}^t, \mathbf{m}^t) = \mathbf{x}^t - \eta_t \mathbf{m}^t$. Since the following discussion focuses on the momentum $\mathbf{m}^t$, we will not specify a fixed choice of $\mathcal{A}_t$.

As a running example, we consider the MARS gradient estimator, which reuses the preceding stochastic gradient and, for $\beta \in (0, 1)$ and $\gamma \in [0, 1]$, takes the following form (Yuan et al., 2024):

$$\mathbf{m}^t = \beta \mathbf{m}^{t-1} + (1-\beta)\left[\nabla f(\mathbf{x}^t; \xi^t) + \gamma \cdot \frac{\beta}{1-\beta}(\nabla f(\mathbf{x}^t; \xi^t) - \nabla f(\mathbf{x}^{t-1}; \xi^{t-1}))\right]. \quad (1)$$

for $\mathbf{m}^0 = \mathbf{0}$ and $\nabla f(\mathbf{x}^{-1}; \xi^{-1}) = \nabla f(\mathbf{x}^0; \xi^0)$. We start with a brief introduction on the technical tools necessary for us to analyze discrete-time recursive filtering systems. For $t = 0, 1, ...$, we will use the notation $x[t]$ to denote an input signal sequence, such as $x[t] = \nabla f(x^t; \xi^t)$, and $y[t]$ to denote a filtering system output, such as $y[t] = \mathbf{m}^t$ the momentum gradient estimator. Throughout this paper, we assume the zero initialization $y[0] = 0$ for all recursive system state variables when not explicitly mentioned. By the example of the momentum in equation 1, we can simplify MARS as a recursive infinite impulse response filtering system $\mathrm{MARS}(\beta, \gamma)$ of

$$y[t] = \beta \cdot y[t] + (1-\beta)\left[x[t] + \gamma \cdot \frac{\beta}{1-\beta}(x[t] - x[t-1])\right]. \quad (2)$$

To illustrate the effect of filtering, we construct a toy example of noisy input

$$x[t] = \sin(2\pi t \cdot \omega_{\text{clean}}) + \sin(2\pi t \cdot \omega_{\text{noise}}) \tag{3}$$

which is the sum of a clean signal in low frequency $\omega_{\text{clean}} = 1/300$ and noise in high frequency $\omega_{\text{noisy}} = 1/10$. We calculate the filter response of MARS filter by equation 2 and illustrate the filtered output by red line in Figure 1. This demonstrates the effect of γ in the MARS filter: as γ increases from 0 to 1, the filtered output transforms from the smooth but delayed signal (top-left figure) to intermediately smoothed with lower delay (top-right figure) and then to unfiltered signal (bottom-right figure).

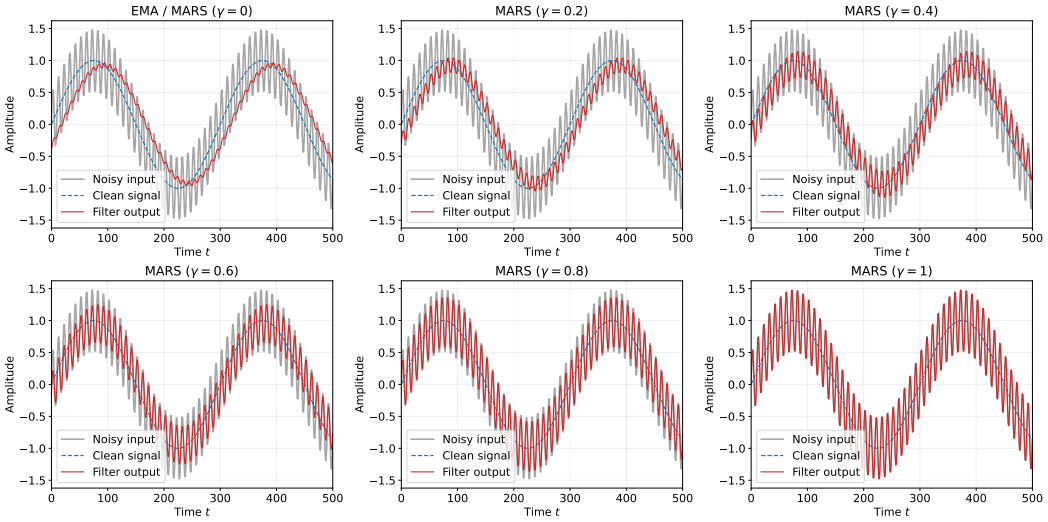

Figure 1: Illustration of filtering a noisy input (grey) with MARS filter defined in equation 2, with different filter configurations of $\gamma \in [0, 1]$ and $\beta = 0.95$. Note that $\gamma = 0$ reduces to the ordinary EMA filter and has the largest delay among all $\gamma \in [0, 1]$.

Now we shall formally define the underlying quantities that govern the above observation. Denote $H(z)$ as the transfer function of a discrete-time filter, where $H(z)$ is obtained by the z-transform of the filter's response function. We denote the amplitude of a sinuous signal of frequency $\omega \in [0, 2\pi]$ at the output of a filter as $G(\omega)$, such that $G(\omega) := |H(e^{i\omega})|$ for $i = \sqrt{-1}$. When $G(\omega)$ is small, the energy level of frequency ω is low at the filter's output. The transfer function also characterize signal delay by the group delay $\tau(\omega) = -\phi'(\omega)$, where $\phi(\omega)$ is the phase response function, i.e., the imaginary exponent of $H(z)$ in $H(e^{i\omega}) = G(\omega)e^{i\phi(\omega)}$. The signal around frequency ω is approximately delayed by $\tau(\omega)$ steps at the output of the filter. Such spectral analysis can be carried out computationally by discrete-time Fourier Transform (DTFT).

The above quantities allow us to characterize the filter response of different existing momentum optimization algorithms, in particular the frequency amplitude response $G(\omega)$ that characterizes stochastic gradient noise filtering and the frequency group delay $\tau(\omega)$ that characterizes gradient bias. In Figure 2, we illustrate $G(\omega)$ and $\tau(\omega)$ for the MARS filter for different configurations of $\gamma \in \{0, 0.025, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Our observation in the previous example can be accurately explained by these two quantities: 1) the group delay $\tau(\omega)$ in the lower frequency band decreases as γ increases, as illustrated by the horizontal time shift of the filter output in Figure 1; 2) the amplitude gain $G(\omega)$ in the higher frequency band increases as γ increases, as illustrated by the high frequency oscillation of the filter output in Figure 1.

Meanwhile, we can perform the same spectral analysis to the other underlying filters that are adopted common optimizers such as Muon (Jordan et al., 2024) and AdEMAMIX (Pagliardini et al., 2025). For instance, Jordan et al. (2024) utilizes a variant of EMA momentum that produces $\mathbf{g}^t$ a linear combination of $\nabla f(\mathbf{x}^t; \xi^t)$ and the ordinary EMA momentum with parameter $\beta \in (0, 1)$:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta \cdot \text{NewtonSchulz}(\mathbf{g}^t), \quad \mathbf{g}^t = \beta^2 \mathbf{m}^{t-1} + (1 - \beta^2)\nabla f(\mathbf{x}^t; \xi^t), \\ \mathbf{m}^t = \beta \mathbf{m}^{t-1} + (1 - \beta)\nabla f(\mathbf{x}^t; \xi^t). \tag{4}$$

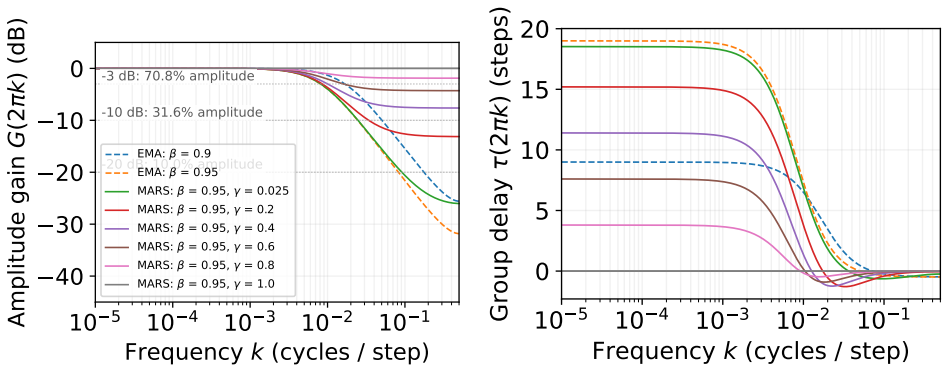

Figure 2: (Left) Amplitude gain $G(2\pi k)$ and (right) group delay $\tau(2\pi k)$ of EMA and MARS filters for frequency $k \in [10^{-5}, 0.5]$. When $\gamma = 0$, the MARS filter reduces to an EMA filter.

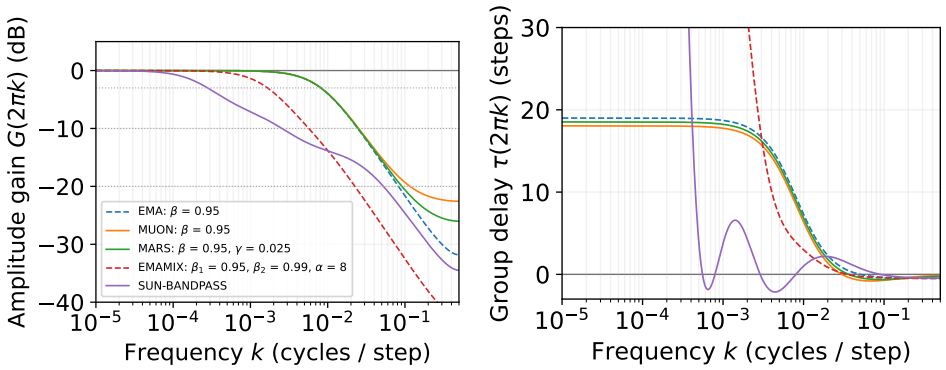

Figure 3: (Left) Amplitude gain $G(2\pi k)$ and (right) group delay $\tau(2\pi k)$ for frequency $k \in [10^{-5}, 0.5]$. The definition of Sun-Bandpass filter can be found in equation 16 and equation 15.

Then, the underlying representing filtering system that produces the gradient estimator $y[t] = \mathbf{g}^t$ with the input stochastic gradient signal $x[t] = \nabla f(\mathbf{x}^t; \xi^t)$ is expressed as

$$y[t] = \beta^2 \cdot z[t-1] + (1-\beta^2)x[t], \quad z[t] = \beta \cdot z[t-1] + (1-\beta)x[t] \tag{5}$$

Similarly, Pagliardini et al. (2025) utilizes a combination of EMA momentum as its gradient estimator $\mathbf{m}^t$ with parameters $\beta_1, \beta_2 \in (0,1), \alpha > 0$:

$$\mathbf{m}^t = \widetilde{\mathbf{m}}_1^t + \alpha \widetilde{\mathbf{m}}_2^t, \ \widetilde{\mathbf{m}}_1^t = \beta_1 \widetilde{\mathbf{m}}_1^t + (1-\beta_1)\nabla f(\mathbf{x}^t; \xi^t), \ \widetilde{\mathbf{m}}_2^t = \beta_2 \widetilde{\mathbf{m}}_2^t + (1-\beta_2)\nabla f(\mathbf{x}^t; \xi^t), \tag{6}$$

and its corresponding filter is expressed as

$$y[t] = z_1[t] + \alpha \cdot z_2[t], \ z_1[t] = \beta_1 \cdot z_1[t-1] + (1-\beta_1) \cdot x[t], \ z_2[t] = \beta_2 \cdot z_2[t-1] + (1-\beta_2) \cdot x[t]. \tag{7}$$

We illustrate their corresponding $G(\omega)$ and $\tau(\omega)$ by Figure 3. We can see that the Muon filter in equation 6 achieves a similar delay reduction effect as the MARS filter. Meanwhile, the EMA mixture filter in equation 7 has significantly larger low frequency group delay.

We remark that some filters have unnormalized DC gain $G(0) > 0$, e.g., the sum of input signal weight coefficients in equation 7 is $1 + \alpha > 1$. For a clearer comparison on the normalized amplitude response, we have rescaled the filter output of equation 7 such that $G(0) = 0$, which is analogously rescaling learning rate η. Note that rescaling does not affect $\tau(\omega)$ by definition. Under these observations, we will illustrate how they connect to the real practical performance later in Section 5.

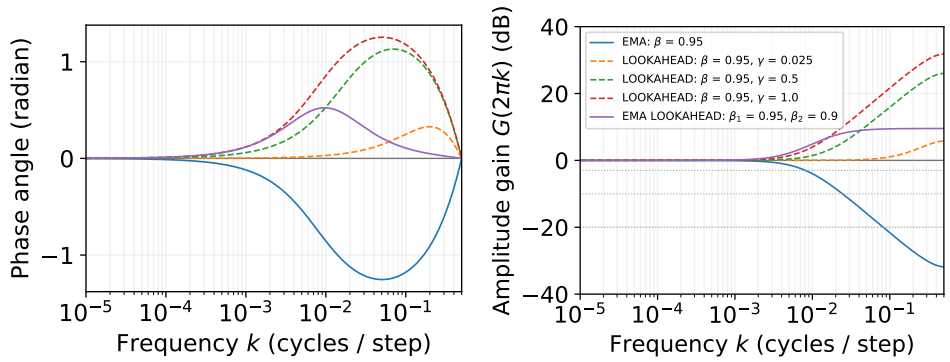

Figure 4: (Left) Phase response $\phi(2\pi k)$ and (right) amplitude gain $G(2\pi k)$ for frequency $k \in [10^{-5}, 0.5]$. Lookahead filter and EMA lookahead filter are defined in equation 8 and equation 9 respectively.

4 Designing Delay Reduced Filters

In this section, we will discuss the optimal filter design for stochastic optimization. Specifically, we want to investigate whether we can construct filtering algorithms with lower signal delay through the means of filter design, as a lower signal delay directly implies a lower gradient bias which is beneficial for training.

4.1 Phase Cancellation by Signal Lookahead

First, we recall the MARS filtering system in equation 2 from the previous section. The γ parameter controls as a step size of a signal lookahead direction $\frac{\beta}{1-\beta} \cdot (x[t] - x[t-1])$. We can immediately notice a physical interpretation of the $\beta/(1-\beta)$ scaling, where the DC group delay of EMA is also $\tau(0) = \beta/(1-\beta)$ (Prandoni & Vetterli, 2008, Sec. 7.3.1). Combining with the fact that $\gamma \to 1$ collapses the MARS filtering system into an identity system, we can tell that the signal lookahead direction added at the input of EMA provides a cancellation effect of phase response, at the cost of weaker high frequency attenuation. This phenomenon is known as a phase compensation. In particular, by analyzing the following lookahead filter in Figure 4,

$$y[t] = x[t] + \gamma \cdot \frac{\beta}{1-\beta}(x[t] - x[t-1]), \tag{8}$$

we can see that it has opposite phase angle and amplitude gain against the EMA filter. γ in equation 8 controls high frequency gain and the high frequency phase angle. When composing the lookahead filter with the EMA filter, both the phase angle and amplitude gain cancel with each other. This illustrates the cancellation of high frequency attenuation and low frequency group delay in MARS as γ increases.

Now in order to solve our toy example filtering problem, we should design a phase compensation filter such that it focuses on phase canceling in the lower frequency band (where our clean signal lives), while avoid altering the higher frequency band to maintain the noise attenuation. This can be achieved by controlling the signal lookahead to pass through only in the lower frequency band, via a low-pass filter such as the EMA: we define the EMA lookahead filter as

$$y[t] = x[t] + \frac{\beta}{1-\beta} \cdot z[t], \quad z[t] = \beta_2 \cdot z[t-1] + (1-\beta_2)(x[t] - x[t-1]). \tag{9}$$

From Figure 4, we can see that the EMA lookahead filter (purple) has the matching lower frequency phase angle with the ideal phase canceling curve (red) until $k = 3 \times 10^{-3}$ which is approximately the clean signal frequency ω_{clean}. Meanwhile, it limits the higher frequency amplitude gain so that the high frequency noise remains attenuated.

Now we define the generalization of the above concept on controlling the signal lookahead direction as the SUN filter: for the purpose of delay reduction in the EMA filter $y[t]$, one can design an internal

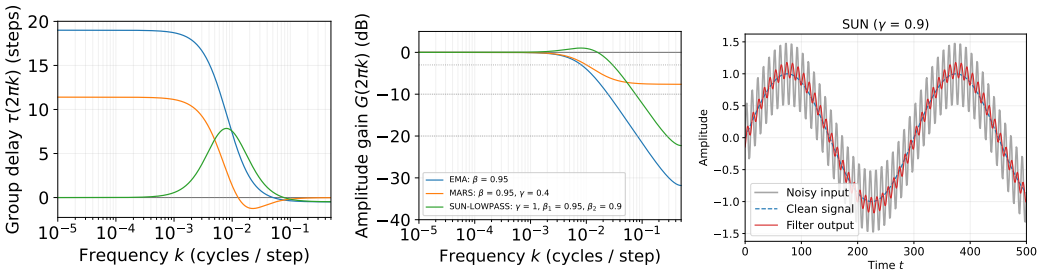

Figure 5: (Left) Group $\tau(2\pi k)$ and (middle) amplitude gain $G(2\pi k)$ for frequency $k \in [10^{-5}, 0.5]$. Lookahead filter and EMA lookahead filter are defined in equation 8 and equation 9 respectively. (Right) Illustration of SUN filter's accurate delay reduced filtering on the toy example.

filtering system $\texttt{FIL}$ on the lookahead signal $\{x[k] - x[k-1]\}_{k=0}^{t}$ and construct a delay reduced EMA filter as

$$y[t] = \beta \cdot y[t] + (1-\beta)\left(x[t] + \frac{\beta}{1-\beta} \cdot \texttt{FIL}(\{x[k] - x[k-1]\}_{k=0}^{t})\right), \tag{10}$$

for $\texttt{FIL}$ an arbitrary FIR/IIR filters, such as the SUN-Lowpass filter in Figure 5 that chooses $\texttt{FIL}$ as EMA as in equation 9. The degree of freedom in choosing $\texttt{FIL}$ determines how accurately the frequency response can be controlled. For instance, when interpreting the SUN-Lowpass filter in equation 9 as a generalization of the MARS filter, an extra degree of freedom on choosing β_2 allows us to control the high frequency gain such that ultimately the SUN-Lowpass filter can accurately filter high frequency noise with zero signal delay, as illustrated in the right figure (red line) in Figure 5.

4.2 Delay Reduced Momentum Algorithms

By the above SUN filtering framework, we can design momentum algorithms that are purposed to have low gradient bias while attenuating high frequency gradient noise. Without loss of generality, we can assume that gradient signal in deep learning are multi-modal and spans a wide frequency band, while stochastic gradient error leans towards higher frequency and largely overlaps with the gradient signal band. An ideal momentum filter should be able to deliver unbiased signal from the gradient signal band while attenuate gradient error frequency band. Under these intuitions, we can aid the momentum design by formulating a constrained optimization problem where the constraints describe the necessary low pass filtering response while the objective function measures the gradient delay. For instance, by choosing $\texttt{FIL}$ as a bandpass filter that is composed by the weighted different of two EMAs, i.e., with parameters $\alpha_1, \alpha_2 > 0$, $\beta, \widetilde{\beta}_1, \widetilde{\beta}_2 \in [0, 1)$ and $\texttt{FIL}(\cdot) = \alpha_1 \texttt{EMA}_{\widetilde{\beta}_1}(\cdot) - \alpha_2 \texttt{EMA}_{\widetilde{\beta}_2}(\cdot)$, we can model our parameterized momentum algorithm SUN-Bandpass as

$$\mathbf{m}^t = \beta \cdot \mathbf{m}^{t-1} + (1-\beta)\left[\nabla f(\mathbf{x}^t; \xi^t) + \frac{\beta}{1-\beta} \cdot (\alpha_1 \cdot \mathbf{h}_1^t - \alpha_2 \cdot \mathbf{h}_2^t)\right], \tag{11a}$$

$$\mathbf{h}_1^t = \widetilde{\beta}_1 \cdot \mathbf{h}_1^{t-1} + (1-\widetilde{\beta}_1)\left[\nabla f(\mathbf{x}^t; \xi^t) - \nabla f(\mathbf{x}^{t-1}; \xi^{t-1})\right], \tag{11b}$$

$$\mathbf{h}_2^t = \widetilde{\beta}_2 \cdot \mathbf{h}_2^{t-1} + (1-\widetilde{\beta}_2)\left[\nabla f(\mathbf{x}^t; \xi^t) - \nabla f(\mathbf{x}^{t-1}; \xi^{t-1})\right]. \tag{11c}$$

for $\mathbf{m}^0 = \mathbf{h}_1^0 = \mathbf{h}_2^0 = \mathbf{0}$ and $\nabla f(\mathbf{x}^{-1}; \xi^{-1}) = \nabla f(\mathbf{x}^0; \xi^0)$. Then, we formulate an optimization problem of minimizing the group delay $|\tau(\omega)|$ over the band $\omega \in [t_0, t_1]$, subject to constraints that characterize the necessary high frequency attenuation $G(\omega)$ over the band $\omega \in [g_0, g_1]$ and the necessary group delay $|\tau(\omega)|$ over the band $\omega \in [k_0, k_1]$:

$$\min_{\substack{\alpha_1,\alpha_2 \geq 0,\\ \beta,\widetilde{\beta}_1,\widetilde{\beta}_2 \in [0,1)}} \quad \max_{\omega \in [t_0,t_1]} \quad |\tau(\omega)| \tag{12}$$

$$\text{s.t.} \quad \underline{G}(\omega) \leq G(\omega) \leq \overline{G}(\omega) \quad \text{for } \omega \in [g_0, g_1], \tag{13}$$

$$\underline{T}(\omega) \leq |\tau(\omega)| \leq \overline{T}(\omega) \quad \text{for } \omega \in [k_0, k_1], \tag{14}$$

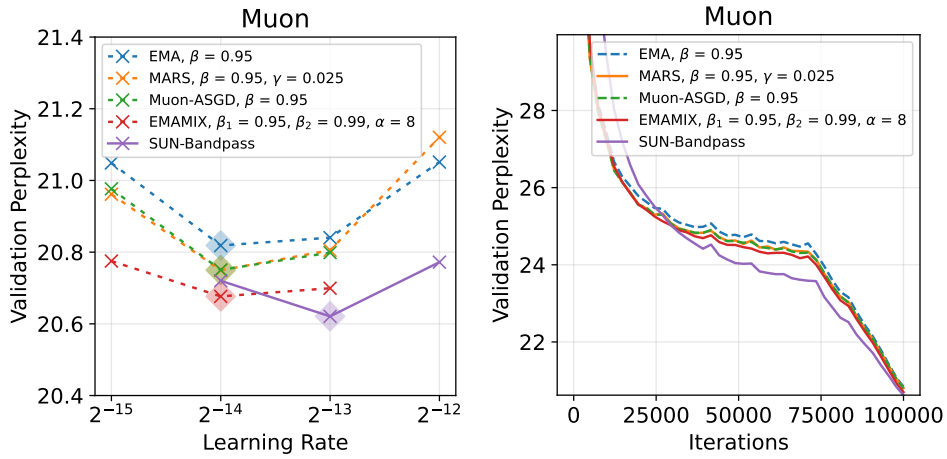

Figure 6: (Left) Learning rate sweeps on logarithmic grid with colored shades highlighting the optimal learning rate. (Right) Learning curves show comparison on the same learning rate 2^{-13}. The actual learning rate is further scaled by 50 for intermediate layers, and by 3,000 for input/output layers.

where constraint functions $\underline{G}(\omega), \overline{G}(\omega), \underline{T}(\omega), \overline{T}(\omega)$ are chosen to help us describe the desired filter response accurately. This non-linearly constrained non-linear optimization problem cannot be solved exactly, while good approximate solutions can be found by optimization solvers such as SLSQP (Kraft, 1988) which iteratively solves local quadratic relaxation of the optimization problem to seek constraint qualification and approximately minimize the objective function. In order to utilize the optimization solver, we shall calculate the analytical form of $\tau(\omega)$ and $G(\omega)$ for the underlying SUN-Bandpass filter in equation 11. Details of the calculation can be found in Appendix A. The objective function and constraints are implemented in an evenly-spaced logarithmic samples of the frequency range.

For the convenience of modeling, we choose $\underline{G}(\omega), \overline{G}(\omega)$ from the function class of $a \cdot \log(\omega) + c$ for some constants $a < 0, c \in \mathbb{R}$ such that they represent straight line boundaries on the amplitude response curve with logarithmic frequency axis. Meanwhile, we choose $\underline{T}(\omega), \overline{T}(\omega)$ from the function class $2\pi r/\omega$ for some constant $r \in (0, 1)$ to ensure that the group delay size is determined relative to the period $2\pi/\omega$.

By iteratively tuning the modeling parameters and executing the SLSQP solver with initialization of previous round results, we are able to discover a SUN-Bandpass filter of coefficients

$$\beta = 0.9876, \quad \widetilde{\beta}_1 = 0.8271, \quad \widetilde{\beta}_2 = 0.9986, \quad \alpha_1 = 0.1698, \quad \alpha_2 = 4.7819 \tag{15}$$

that demonstrates a low-pass behavior with a better controlled mid-frequency group delay, as illustrated in Figure 3. We remark that the above solution is not guaranteed to be the optimal coefficients, and is rather an approximate solution that can be further refined. We will demonstrate the practical training behavior of gradient estimation under the above SUN-Bandpass momentum.

5 Numerical Experiments

Our numerical experiments are based on the NanoGPT[1] language model with 124M model parameters, and performing pre-training with a batch size of 0.5M tokens (sequence length of 1,024 tokens) using the FineWeb dataset (Penedo et al., 2024). We use a learning rate under constant-decay schedule, where the learning rate is constant for the first 71,570 steps and decay to 0 for the remaining steps, over the course of 100,000 iterations. Model performance is evaluated on a held-out validation dataset of 10M tokens. Our experiments are performed on GPU servers of 8× NVIDIA H200.

For optimizer $\mathcal{A}_t$, we use the Muon optimizer (Jordan et al., 2024) and replace its momentum with other baseline momentum algorithms including EMA, MARS (Liu et al., 2025b), EMAMIX

[1] https://github.com/KellerJordan/modded-nanogpt/

Momentum Algorithm	Validation perplexity
EMA	20.819
MARS	20.749
Muon-ASGD	20.750
EMAMIX	20.677
SUN-Bandpass	**20.621**

Table 1: Validation perplexity of the last iteration for each momentum algorithm under the optimal learning rate searched in 6.

(Pagliardini et al., 2025) as defined in equation 7 and the original Muon's momentum (Muon-ASGD) as defined in equation 6. Specifically, we follow Liu et al. (2025b) and keep the input/output layer optimizer as Adam (Kingma & Ba, 2014) as is implemented in the Muon algorithm, and only replace the momentum in the intermediate layers.

By the results in Figure 6 and Table 1, we can see that the performance of different momentum algorithms matches with the ordering of group delay magnitude in Figure 3. Specifically, MARS and Muon-ASGD show a slight reduction in the mid-frequency group delay in Figure 3, and it is reflected in a slight acceleration on top of EMA momentum in Figure 6. Meanwhile, by a significant mid-frequency group delay gap between EMAMIX and EMA, it explains the larger performance gain of EMAMIX among other baselines. Our SUN-Bandpass filter is designed to have minimal group delay at the mid-frequency as illustrated in Figure 3, and it naturally translates to the largest performance gain in Figure 6 of -0.198 validation perplexity drop compared to the ordinary EMA.

6 Conclusion

In this paper, we take on the filter analysis perspective and interpret momentum algorithms as filtering systems with signal delay. The analysis on signal delay enables us to quantify gradient bias as the group delay of stochastic gradient signal. This approach provides a transparent interpretation about the gradient bias of momentum, without depending on technical assumptions that are convoluted in existing convergence analysis. Our result complements with the existing analysis on momentum algorithms and provides a novel insight on the practical success of momentum variance reduction algorithms (STORM, MARS) in a broad sense. Besides the analytical understanding, our framework stochastic unbiased momentum (SUN) enables tools from filter design to guide the choice of practical momentum algorithms, leading to momentum algorithms with minimal gradient delay and practical acceleration in language model pre-training experiments. We foresee that future research direction along this path will combine existing filter design techniques into momentum algorithm design.

AI use statement

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for polishing paper writing, research idea brain-storming and implementation, routine research operations and literature survey. Specifically, the language of writing is improved by AI and we have verified that the AI output sentences that carry the same intended meaning as the original sentence. During the development of this project, AI is in the process of formulating mathematical equations and calculating the analytical equations in Appendix 3. We have verified the correctness of the mathematical derivation using AI and manual checking. The AI is also used as a literature search engine to identify important papers on this topic. We have reviewed all AI-assisted work for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Ricky T. Q. Chen, Dami Choi, Lukas Balles, David Duvenaud, and Philipp Hennig. Self-tuning stochastic optimization with curvature-aware gradient filtering. In *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning*

Research, pp. 60–69. PMLR, 2020. URL https://proceedings.mlr.press/v137/chen20a.html.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/1543843a4723ed2ab08e18053ae6dc5b-Abstract.html.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. In *Conference On Learning Theory*, pp. 545–604. PMLR, 2018.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Abstract.html.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024.

Rahul Kidambi, Praneeth Netrapalli, Prateek Jain, and Sham Kakade. On the insufficiency of existing momentum schemes for stochastic optimization. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9. IEEE, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dieter Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, DFVLR, 1988.

Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016. doi: 10.1137/15M1009597.

Xianliang Li, Jun Luo, Zhiwei Zheng, Hanxiao Wang, Li Luo, Lingkun Wen, Linlong Wu, and Sheng Xu. On the performance analysis of momentum method: A frequency domain perspective. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/76c64372d25513fed60a9e4176a62481-Paper-Conference.pdf.

Xianliang Li, Zihan Zhang, Weiyang Liu, and Han Bao. Denoise first, orthogonalize later: Understanding momentum in muon via spectral filtering. *arXiv preprint arXiv:2606.03899*, 2026.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.14342.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025a.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. *Advances in neural information processing systems*, 33:18261–18271, 2020.

Yifeng Liu, Angela Yuan, and Quanquan Gu. Mars-m: When variance reduction meets matrices. *arXiv preprint arXiv:2510.21800*, 2025b.

Hesameddin Mohammadi, Meisam Razaviyayn, and Mihailo R. Jovanović. Robustness of accelerated first-order algorithms for strongly convex optimization problems. *arXiv preprint arXiv:1905.11011*, 2019. URL https://arxiv.org/abs/1905.11011.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. In *Dokl akad nauk Sssr*, volume 269, pp. 543, 1983.

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2613–2621. PMLR, 2017. URL https://proceedings.mlr.press/v70/nguyen17b.html.

Matteo Pagliardini, Pierre Ablin, and David Grangier. The ademamix optimizer: Better, faster, older. In *International Conference on Learning Representations*, volume 2025, pp. 64715–64757, 2025.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Paolo Prandoni and Martin Vetterli. *Signal Processing for Communications*. EPFL Press, Lausanne, Switzerland, 2008. URL https://www.sp4comm.org/webversion/livre.html.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 2024. URL https://arxiv.org/abs/2409.11321.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025. URL https://arxiv.org/abs/2509.02046.

Huizhuo Yuan, Yifeng Liu, Shuang Wu, Xun Zhou, and Quanquan Gu. Mars: Unleashing the power of variance reduction for training large models. *arXiv preprint arXiv:2411.10438*, 2024.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham Kakade. Deconstructing what makes a good optimizer for language models. *arXiv preprint arXiv:2407.07972*, 2024. URL https://arxiv.org/abs/2407.07972.

A Analytical Form of Group Delay and Amplitude Gain for SUN-Bandpass Filter

The SUN-Bandpass filtering system that is used in equation 11 can be described as a discrete-time filter of internal states $z_1[t], z_2[t]$ that produces output $y[t]$ when given input signal $x[t]$ such that for $t = 0, 1, 2, ...,$

$$y[t] = \beta \cdot y[t-1] + (1-\beta)\left(x[t] + \frac{\beta}{1-\beta}(\alpha_1 \cdot z_1[t] - \alpha_2 \cdot z_2[t])\right), \tag{16a}$$

$$z_1[t] = \widetilde{\beta}_1 \cdot z_1[t-1] + (1-\widetilde{\beta}_1)\,(x[t] - x[t-1])\,, \tag{16b}$$

$$z_2[t] = \widetilde{\beta}_2 \cdot z_2[t-1] + (1-\widetilde{\beta}_2)\,(x[t] - x[t-1])\,, \tag{16c}$$

for $\alpha_1, \alpha_2 > 0$, $\beta, \widetilde{\beta}_1, \widetilde{\beta}_2 \in [0, 1)$, $y[0] = z_1[0] = z_2[0] = 0$ and $x[-1] = x[0]$.

Define the nonnegative auxiliary parameters

$$\lambda_0 = \frac{\beta}{1-\beta}, \qquad \lambda_1 = \frac{\widetilde{\beta}_1}{1-\widetilde{\beta}_1}, \qquad \lambda_2 = \frac{\widetilde{\beta}_2}{1-\widetilde{\beta}_2}. \tag{17}$$

Angular frequency $\omega \in [0, 2\pi]$ is measured in radians per update.

1. Derivation of the transfer function. For an EMA with rate r, the zero-state recurrence $v[t] = rv[t-1] + (1-r)u[t]$ gives, upon taking the z-transform,

$$V(z) = rz^{-1}V(z) + (1-r)U(z), \qquad \frac{V(z)}{U(z)} = \frac{1-r}{1-rz^{-1}}.$$

Denote this transfer function by

$$E_r(z) = \frac{1-r}{1-rz^{-1}} = \frac{1}{1+\frac{r}{1-r}(1-z^{-1})}. \tag{18}$$

The z-transform of $x[t] - x[t-1]$ is $(1-z^{-1})X(z)$. Writing $Z_j(z)$ for the transform of the branch sequence $z_j[t]$, the branch recurrences therefore give

$$Z_j(z) = \widetilde{\beta}_j z^{-1} Z_j(z) + (1-\widetilde{\beta}_j)(1-z^{-1})X(z), \qquad j = 1, 2,$$

and hence

$$\frac{Z_j(z)}{X(z)} = \frac{(1-\widetilde{\beta}_j)(1-z^{-1})}{1-\widetilde{\beta}_j z^{-1}} = (1-z^{-1})E_{\widetilde{\beta}_j}(z).$$

The outer recurrence in equation 16 gives

$$Y(z) = E_\beta(z)\left[X(z) + \lambda_0\left(\alpha_1 Z_1(z) - \alpha_2 Z_2(z)\right)\right].$$

Dividing by $X(z)$ and denoting the resulting transfer function by $H(z) = Y(z)/X(z)$ yields

$$H(z) = E_\beta(z)\left[1 + \lambda_0(1-z^{-1})\left(\alpha_1 E_{\widetilde{\beta}_1}(z) - \alpha_2 E_{\widetilde{\beta}_2}(z)\right)\right]. \tag{19}$$

Substituting the EMA transfer functions gives

$$H(z) = \frac{1}{1+\lambda_0(1-z^{-1})}\left[1 + \lambda_0(1-z^{-1})\left(\frac{\alpha_1}{1+\lambda_1(1-z^{-1})} - \frac{\alpha_2}{1+\lambda_2(1-z^{-1})}\right)\right]. \tag{20}$$

Evaluate on $z = e^{i\omega}$ and define $s(\omega) := 1 - e^{-i\omega}$. Combining the fractions gives

$$H(e^{i\omega}) = \frac{(1+\lambda_1 s(\omega))(1+\lambda_2 s(\omega)) + \lambda_0 s(\omega)\left[\alpha_1(1+\lambda_2 s(\omega)) - \alpha_2(1+\lambda_1 s(\omega))\right]}{(1+\lambda_0 s(\omega))(1+\lambda_1 s(\omega))(1+\lambda_2 s(\omega))} \tag{21}$$

$$= \frac{1 + \left[\lambda_1 + \lambda_2 + \lambda_0(\alpha_1 - \alpha_2)\right] s(\omega) + \left[\lambda_1\lambda_2 + \lambda_0(\alpha_1\lambda_2 - \alpha_2\lambda_1)\right] s(\omega)^2}{\prod_{j=0}^{2}(1+\lambda_j s(\omega))}. \tag{22}$$

The expansion identifies the coefficient of $s(\omega)$ and the coefficient of $s(\omega)^2$. Now we simplify these coefficients as

$$n_1 = \lambda_1 + \lambda_2 + \lambda_0(\alpha_1 - \alpha_2), \tag{23}$$
$$n_2 = \lambda_1\lambda_2 + \lambda_0(\alpha_1\lambda_2 - \alpha_2\lambda_1). \tag{24}$$

Thus the numerator of $H(e^{i\omega})$ is simplified as

$$N(s(\omega)) = 1 + n_1 s(\omega) + n_2 s(\omega)^2. \tag{25}$$

The frequency response can therefore be written as

$$H(e^{i\omega}) = \frac{1 + n_1 s(\omega) + n_2 s(\omega)^2}{\prod_{j=0}^{2}(1 + \lambda_j s(\omega))} = \frac{N(s(\omega))}{\prod_{j=0}^{2}(1 + \lambda_j s(\omega))}, \qquad s(\omega) = 1 - e^{-i\omega}. \tag{26}$$

2. Derivation of the amplitude gain. The linear amplitude gain is the magnitude of the frequency response. Taking absolute values in equation 26 gives

$$G(\omega) := |H(e^{i\omega})| = \frac{|1 + n_1 s(\omega) + n_2 s(\omega)^2|}{\prod_{j=0}^{2} |1 + \lambda_j s(\omega)|}. \tag{27}$$

Since $s(\omega) = 1 - \cos\omega + i\sin\omega$, each denominator factor satisfies

$$|1 + \lambda_j s(\omega)|^2 = [1 + \lambda_j(1 - \cos\omega)]^2 + \lambda_j^2 \sin^2\omega \tag{28}$$
$$= 1 + 2\lambda_j(1 + \lambda_j)(1 - \cos\omega). \tag{29}$$

Combining the above yields the analytical amplitude gain

$$\boxed{G(\omega) = \frac{|1 + n_1 s(\omega) + n_2 s(\omega)^2|}{\prod_{j=0}^{2} \sqrt{1 + 2\lambda_j(1 + \lambda_j)(1 - \cos\omega)}}.} \tag{30}$$

3. Derivation of the group delay. Throughout this derivation, a prime denotes differentiation with respect to a function's argument. To distinguish differentiation with respect to angular frequency from differentiation with respect to the transform variable, write

$$\mathcal{H}(\omega) := H(e^{i\omega}). \tag{31}$$

For the delay, choose a differentiable local phase $\phi(\omega)$ at a frequency where $\mathcal{H}(\omega) \neq 0$. Differentiating $\mathcal{H}(\omega) = G(\omega)e^{i\phi(\omega)}$ gives

$$\mathcal{H}'(\omega) = G'(\omega)e^{i\phi(\omega)} + iG(\omega)\phi'(\omega)e^{i\phi(\omega)}. \tag{32}$$

Dividing by $\mathcal{H}(\omega)$ yields

$$\frac{\mathcal{H}'(\omega)}{\mathcal{H}(\omega)} = \frac{G'(\omega)}{G(\omega)} + i\phi'(\omega), \qquad \tau(\omega) = -\phi'(\omega) = -\operatorname{Im}\frac{\mathcal{H}'(\omega)}{\mathcal{H}(\omega)}. \tag{33}$$

To evaluate this derivative explicitly, first differentiate $s(\omega) = 1 - e^{-i\omega}$:

$$s'(\omega) = ie^{-i\omega}. \tag{34}$$

Differentiating the numerator polynomial with respect to its argument ξ and then evaluating at $s(\omega)$ gives

$$N'(\xi) = n_1 + 2n_2\xi, \qquad N'(s(\omega)) = n_1 + 2n_2 s(\omega). \tag{35}$$

Each denominator factor $1 + \lambda_j s(\omega)$ has derivative $\lambda_j s'(\omega)$ with respect to ω. The logarithmic derivative of a quotient is the numerator's logarithmic derivative minus the denominator's logarithmic derivative. For the product in the denominator, the latter is the sum over its three factors. Hence equation 26, together with the chain rule, gives

$$\frac{\mathcal{H}'(\omega)}{\mathcal{H}(\omega)} = s'(\omega)\left[\frac{N'(s(\omega))}{N(s(\omega))} - \sum_{j=0}^{2}\frac{\lambda_j}{1 + \lambda_j s(\omega)}\right] \tag{36}$$

$$= ie^{-i\omega}\left[\frac{n_1 + 2n_2 s(\omega)}{1 + n_1 s(\omega) + n_2 s(\omega)^2} - \sum_{j=0}^{2}\frac{\lambda_j}{1 + \lambda_j s(\omega)}\right]. \tag{37}$$

Taking the negative imaginary part therefore gives the analytical signed group delay

$$\boxed{\tau(\omega) = -\operatorname{Im}\left\{ ie^{-i\omega} \left[\frac{n_1 + 2n_2 s(\omega)}{1 + n_1 s(\omega) + n_2 s(\omega)^2} - \sum_{j=0}^{2} \frac{\lambda_j}{1 + \lambda_j s(\omega)} \right] \right\}.} \tag{38}$$

This formula applies wherever $N(s(\omega)) \neq 0$. The denominator factors cannot vanish on the unit circle because the EMA rates are less than one.