VENUS: DECOUPLING THE SAMPLES IN MARS BENEFITS SMALL-BATCH TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

AdamW has long been a standard optimizer for neural network training, with recent alternatives including matrix-aware methods such as Muon and variance-reduced methods such as MARS, which scales the variance-reduction correction term. We show that the unclipped, estimator-level formulation of MARS is exactly a convex combination of momentum SGD (MSGD) and momentum variance reduction (MVR). When both branches use the same minibatch, the three mixing parameters reduce to two effective MARS parameters; thus, varying these parameters independently does not expand the estimator family. Independent sampling, however, does. We introduce VENUS, which evaluates the two branches on independent minibatches and introduces one additional effective degree of freedom. We establish convergence guarantees for updates defined by a linear minimization oracle (LMO), where the additional mixing weight enters through an explicit variance factor minimized by balanced weighting. Experiments on GPT-2-style language model pretraining under matched gradient budgets support the same balanced weighting and reveal a batch-size-dependent pattern: exact VENUS achieves lower final validation loss at batch size 16, the difference at 32 remains inconclusive, and exact MARS achieves lower loss at 64. In the tested cached-gradient configurations at batch sizes 16 and 32, cached MARS achieves lower final validation loss than cached VENUS and both exact estimators. Fixed-state analyses separate independent-gradient averaging from refresh–transport covariance and show how covariance shifts the variance-minimizing split. Diagnostic experiments further distinguish estimator variance from optimization progress: the equal-work comparison favors different estimators under Euclidean and layerwise LMO updates. The observed small-batch benefit of exact sample decoupling motivates further study of training under limited computational resources.

1 INTRODUCTION

Training a modern neural network is an instance of stochastic nonconvex optimization,

$$\min_{x\in\mathbb{R}^d} \; f(x) \;:=\; \mathbb{E}_{\xi\sim\mathcal{D}}\left[f_\xi(x)\right], \tag{1}$$

where ξ is a data sample drawn from an unknown distribution $\mathcal{D}$, each f_ξ is possibly nonconvex, and d is large.

Language models have scaled from GPT-2 and GPT-3 (Radford et al., 2019; Brown et al., 2020) to frontier systems such as GPT-5 (Bubeck et al., 2025), Llama 3 (Grattafiori et al., 2024) and DeepSeek-R1 (Guo et al., 2025), which comprise billions of parameters and are trained on massive corpora. Throughout that scale-up, AdamW and its relatives (Kingma & Ba, 2015; Loshchilov & Hutter, 2017) have remained the standard optimizers for (1) and the default baselines for evaluating new optimizers.

Two developments have begun to challenge that dominance. **The first** exploits the matrix or tensor structure of layer parameters rather than treating them as flat vectors. Muon (Jordan et al., 2024b) is an early method in this direction: it approximately orthogonalizes an SGD-momentum update using Newton–Schulz iterations and outperforms AdamW on language-model training tasks. Liu et al. (2025a) demonstrate that the method scales to distributed LLM pretraining. Related structure-aware approaches include Shampoo (Gupta et al., 2018) and SOAP (Vyas et al., 2025). Pethick et al. (2025)

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.

interpret the idealized Muon update (Jordan et al., 2024b) as a *linear minimization oracle* (LMO) over a spectral-norm ball and write the update as

$$x_{k+1} = x_k + s_k, \qquad s_k \in \arg\min_{\|s\| \le \eta_k} \langle m_k, s \rangle. \tag{2}$$

For each m, choose $\mathrm{lmo}_\eta(m) \in \arg\min_{\|s\| \le \eta} \langle m, s \rangle$. Then $x_{k+1} = x_k + \mathrm{lmo}_{\eta_k}(m_k)$, where the norm determines the update geometry and m_k is a stochastic gradient surrogate. In all algorithms below, we hold the LMO geometry fixed and vary only the construction of m_k. A basic reference estimator is the exponential moving average used by momentum SGD, which we denote by MSGD:

$$m_k^{\mathrm{MSGD}}(\mu) := (1-\mu) m_{k-1} + \mu \nabla f_{\xi_k}(x_k). \tag{3}$$

The second development is a more effective use of variance reduction. Momentum variance reduction (MVR) (Cutkosky & Orabona, 2019) augments the momentum recursion with a stochastic gradient difference,

$$m_k^{\mathrm{MVR}}(p) := (1-p)\big(m_{k-1} + \nabla f_{\xi_k}(x_k) - \nabla f_{\xi_k}(x_{k-1})\big) + p \nabla f_{\xi_k}(x_k), \tag{4}$$

which improves the rate in the gradient norm from $\mathcal{O}(T^{-1/4})$ to $\mathcal{O}(T^{-1/3})$. Naive applications of classical variance-reduction methods such as SVRG had shown limited practical benefit on modern deep networks (Defazio & Bottou, 2019); MARS (Yuan et al., 2025) renewed interest in this direction by *scaling* the correction. Its fixed-γ, unclipped estimator-level form is

$$m_k^{\mathrm{MARS}}(\beta, \gamma) := (1-\beta)\big(m_{k-1} + \gamma(\nabla f_{\xi_k}(x_k) - \nabla f_{\xi_k}(x_{k-1}))\big) + \beta \nabla f_{\xi_k}(x_k), \tag{5}$$

which recovers MSGD at $\gamma = 0$ and MVR at $\gamma = 1$ (Yuan et al., 2025; Shulgin et al., 2026), i.e. $m_k^{\mathrm{MARS}}(\beta, 0) = m_k^{\mathrm{MSGD}}(\beta)$ and $m_k^{\mathrm{MARS}}(\beta, 1) = m_k^{\mathrm{MVR}}(\beta)$. [1] Shulgin et al. (2026) explain why an intermediate scale can help, showing that it can make the governing similarity constant strictly smaller than at either endpoint, and MARS-M (Liu et al., 2025b) extends the construction to the Moonlight variant of Muon.

Independent sampling in hybrid estimators is not itself new: the hybrid SARAH–SGD methods of Tran-Dinh et al. (2019; 2022) use independent samples, whereas Liu et al. (2020) share them to reduce oracle cost. These works neither compare coupled and independent sampling under a matched gradient budget nor consider MARS-style correction scaling, matrix-aware geometry, or large-model training. We revisit this trade for MARS in the LMO setting.

1.1 Contributions

(i) We recast the MARS estimator (5) as a convex combination of MSGD and MVR (Proposition 1), thereby making the full mixture explicit, whereas only its endpoints were previously on record (Yuan et al., 2025; Shulgin et al., 2026). The same proposition also identifies the obstruction: when the two candidates share the minibatch ξ_k, the three scalar parameters (θ, μ, p) collapse onto the two effective MARS parameters (β, γ). Thus, untying the scalar weights alone cannot leave the MARS family; the samples themselves must be decoupled.

(ii) Motivated by this obstruction, we introduce VENUS (Algorithm 2), a convex combination of MSGD and MVR branches evaluated on *independent* minibatches. Its effective parametrization introduces an additional sample-splitting weight τ, which controls the independent fresh-gradient contribution, with coupled MARS recovered at $\tau = 0$. We establish convergence guarantees for LMO updates defined by a general norm (Theorem 1), with an explicit split-dependent factor $\chi_\tau = \sqrt{\tau^2 + (1-\tau)^2}$ minimized at $\tau = 0.5$. Performance at matched gradient budgets is regime-dependent. On CIFAR-10, VENUS achieves the best mean summary metrics in the equal-gradients-per-step regime, compared to MARS and AdamW. In GPT-2-style pretraining, exact VENUS has lower final validation loss on all six confirmation seeds at batch size 16, exact MARS does so at 64, and the difference at 32 is inconclusive (Section 5). At batch sizes 16 and 32, both tested cached-gradient variants outperform both exact estimators, with cached MARS achieving the lowest loss (Appendix D.5). Controlled experiments distinguish the effects of independent-gradient averaging

[1] Here and below we display an estimator's parameters, as in $m_k^{\mathrm{MARS}}(\beta, \gamma)$, only where the dependence on them is at issue; in the pseudocode the parameters are fixed by the algorithm's input, so we write m_k and m_k^{MSGD} without them.

Algorithm 1 MARS

Require: x_0, m_0, horizon K, stepsizes $\{\eta_k\}_{k=0}^K$, momentum weight $\beta \in (0,1]$, MVR scale $\gamma \in [0,1]$

for $k = 0,1,\ldots,K$ **do**
  $x_{k+1} \leftarrow x_k + \mathrm{lmo}_{\eta_k}(m_k)$
  sample one minibatch ξ_{k+1}
  $g^+_{k+1} \leftarrow \nabla f_{\xi_{k+1}}(x_{k+1}), \quad g^0_{k+1} \leftarrow \nabla f_{\xi_{k+1}}(x_k)$ (both on the *same* batch)
  $m_{k+1} \leftarrow (1-\beta)\big(m_k + \gamma(g^+_{k+1} - g^0_{k+1})\big) + \beta g^+_{k+1}$
end for

Algorithm 2 VENUS

Require: x_0, m_0, horizon K, stepsizes $\{\eta_k\}_{k=0}^K$, refresh weight $\beta \in (0,1)$, correction scale $\gamma \in [0,1]$, sample-splitting weight $\tau \in [0,1]$

for $k = 0,1,\ldots,K$ **do**
  $x_{k+1} \leftarrow x_k + \mathrm{lmo}_{\eta_k}(m_k)$
  sample two fresh, conditionally independent minibatches ξ_{k+1} and ζ_{k+1}
  $h^+_{k+1} \leftarrow \nabla f_{\xi_{k+1}}(x_{k+1})$
  $g^+_{k+1} \leftarrow \nabla f_{\zeta_{k+1}}(x_{k+1}), \quad g^0_{k+1} \leftarrow \nabla f_{\zeta_{k+1}}(x_k)$ (same ζ_{k+1})
  $m_{k+1} \leftarrow (1-\beta)\big(m_k + \gamma(g^+_{k+1} - g^0_{k+1})\big) + \beta\big(\tau h^+_{k+1} + (1-\tau)g^+_{k+1}\big)$
end for

and refresh–transport covariance on estimator variance. At fixed CNN states, VENUS with $\tau = 0.5$ improves mean one-step objective progress over larger-batch MARS at equal gradient work under the layerwise LMO. Transformer diagnostics support variance-minimizing splits near 0.5(Section 6). The small-batch gains of exact VENUS motivate further study of sample decoupling for training under limited computational resources.

2 MARS IS A CONVEX COMBINATION OF MSGD AND MVR

In this section, we isolate the algebraic observation that motivates the VENUS method introduced below. As noted by Shulgin et al. (2026), the estimator (5) reduces to MSGD at $\gamma = 0$ and to MVR at $\gamma = 1$. The next result strengthens these two endpoints into an identity along the whole segment: whenever the two candidates read the same minibatch, MARS is *exactly* a convex combination of MSGD and MVR. Throughout, we write $\mathrm{Conv}_\theta(a,b) := (1-\theta)a + \theta b$.

Proposition 1 (Shared-sample MARS decomposition)**.** *Fix an incoming estimate m_{k-1} and evaluate the MSGD and MVR candidates in* (3) *and* (4) *on the same minibatch ξ_k. Let $\theta \in [0,1]$, $\mu \in (0,1]$ and $p \in (0,1)$, and define*

$$\beta := (1-\theta)\mu + \theta p. \tag{6}$$

Suppose that $\beta < 1$, and define

$$\gamma := \frac{\theta(1-p)}{1-\beta}. \tag{7}$$

Then $\gamma \in [0,1]$ and

$$\mathrm{Conv}_\theta\big(m_k^{\mathrm{MSGD}}(\mu), m_k^{\mathrm{MVR}}(p)\big) = m_k^{\mathrm{MARS}}(\beta,\gamma). \tag{8}$$

Conversely, every MARS estimator with $\beta \in (0,1)$ and $\gamma \in [0,1]$ arises in this way, with $\mu = p = \beta$ and $\theta = \gamma$. At the boundary $\beta = 1$, the MARS estimator reduces to $\nabla f_{\xi_k}(x_k)$ independently of γ and is recovered by taking $\theta = 0$ and $\mu = 1$; the effective correction scale is then non-identifiable.

The proof is given in Appendix A. Beyond making the mixture explicit, (6)–(7) carry an immediate negative consequence: under shared sampling, the three scalar parameters (θ, μ, p) collapse onto the two-parameter interpolation family of MARS, with $\beta = 1$ understood as a degenerate boundary at which γ is immaterial. Untying the scalar weights therefore cannot leave this MARS family. The extra degree of freedom exploited in Section 3.1 comes from untying the *samples*, not the scalars.

3 DECOUPLING THE SAMPLES GIVES A NEW METHOD: VENUS

3.1 ALGORITHM CONSTRUCTION

Proposition 1 shows that shared sampling restricts the mixture to the MARS family. VENUS removes this restriction by evaluating the MSGD and MVR branches on fresh, conditionally independent minibatches ξ_{k+1} and ζ_{k+1}, respectively:

$$m_{k+1}^{\mathrm{MSGD}}(\mu) = (1-\mu)m_k + \mu\nabla f_{\xi_{k+1}}(x_{k+1}),$$
$$m_{k+1}^{\mathrm{MVR}}(p) = p\nabla f_{\zeta_{k+1}}(x_{k+1}) + (1-p)\big[m_k + \nabla f_{\zeta_{k+1}}(x_{k+1}) - \nabla f_{\zeta_{k+1}}(x_k)\big]. \tag{9}$$

Both branches use the same incoming estimate m_k, and their outputs are mixed with weight θ. Retaining β and γ from (6)–(7), with $0 < \beta < 1$, define the sample-splitting weight

$$\tau := \frac{(1-\theta)\mu}{\beta} \in [0,1]. \tag{10}$$

The resulting estimator is

$$\begin{aligned} m_{k+1}^{\mathrm{VENUS}}(\beta,\gamma,\tau) &:= (1-\theta)m_{k+1}^{\mathrm{MSGD}}(\mu) + \theta m_{k+1}^{\mathrm{MVR}}(p) \\ &= (1-\beta)\Big(m_k + \gamma\big[\nabla f_{\zeta_{k+1}}(x_{k+1}) - \nabla f_{\zeta_{k+1}}(x_k)\big]\Big) \\ &\quad + \beta\Big[\tau\nabla f_{\xi_{k+1}}(x_{k+1}) + (1-\tau)\nabla f_{\zeta_{k+1}}(x_{k+1})\Big]. \end{aligned} \tag{11}$$

Here β controls the *refresh* contribution from current-point gradients, γ scales the same-sample *transport* difference, and τ allocates the refresh weight to the independent ξ gradient. Algorithm 2 implements this effective form, which is also the parametrization used for tuning in our main LLM study. Every $(\beta,\gamma,\tau) \in (0,1)\times[0,1]^2$ corresponds to a branch mixture in the continuous closure of the primitive parametrization; Appendix B.1 gives the inverse map.

This construction extends MARS in a precise sense. Setting $\tau = 0$ recovers Algorithm 1, with the unused ξ gradient omitted. Alternatively, identifying $\xi_{k+1} = \zeta_{k+1}$ makes the estimator independent of τ and recovers MARS for any splitting weight. With independent samples, the two refresh gradients have separate coefficients, so τ remains an additional effective degree of freedom at fixed (β,γ). The general VENUS update requires three gradient evaluations rather than two; unused evaluations can be omitted at boundary settings. In our experiments we compare VENUS and MARS at matched gradient budgets in the LLM and CNN studies (Sections 5 and 4).

3.2 CONVERGENCE ANALYSIS

3.2.1 ASSUMPTIONS

Let $\|\cdot\|$ be the norm defining the LMO ball and $\|\cdot\|_*$ its dual. We assume that $f_{\inf} := \inf_x f(x) > -\infty$ and that $\|\nabla f(x) - \nabla f(y)\|_* \le L\|x-y\|$ for all $x,y \in \mathbb{R}^d$. The stochastic gradients are unbiased, $\mathbb{E}_\xi[\nabla f_\xi(x)] = \nabla f(x)$, and have bounded variance in the Euclidean norm, $\mathbb{E}_\xi[\|\nabla f_\xi(x) - \nabla f(x)\|_2^2] \le \sigma^2$. At each iteration, the samples ξ_{k+1} and ζ_{k+1} are fresh and conditionally independent. For $d_\xi(x,y) := \nabla f_\xi(x) - \nabla f_\xi(y)$, $d(x,y) := \nabla f(x) - \nabla f(y)$, following Shulgin et al. (2026), we assume, for the fixed effective correction scale $\gamma \in [0,1]$, that

$$\mathbb{E}_\xi\big[\|\gamma d_\xi(x,y) - d(x,y)\|_*^2\big] \le \delta_\gamma^2\|x-y\|^2, \qquad x,y \in \mathbb{R}^d. \tag{A}$$

For a general LMO geometry, let $\rho\|v\|_2 \le \|v\|_* \le \bar\rho\|v\|_2$ and define $\sigma_* := \bar\rho\sigma, \delta_{\gamma,*} := \frac{\bar\rho}{\rho}\delta_\gamma$. In Euclidean geometry, $\rho = \bar\rho = 1$.

Condition (A) is the γ-similarity condition of Shulgin et al. (2026); it reduces to standard mean-squared similarity at $\gamma = 1$ and follows from L-smoothness at $\gamma = 0$, so it imposes no stronger regularity than either. A smaller δ_γ is accompanied, however, by the $(1-\gamma)$-dependent drift term in Theorem 1. A fuller discussion, including the admissible choices of δ_γ, is deferred to Appendix A.1.

3.2.2 COMPLEXITY RESULTS

Let $N := K + 1$, let $\eta_k \equiv \eta$, and let R, independently of the algorithmic randomness, be uniform on $\{0, \dots, N-1\}$. Define $\chi_\tau := \sqrt{\tau^2 + (1-\tau)^2}$.

Theorem 1 (Complexity of VENUS). *Suppose the assumptions of Section 3.2.1 hold and let $(\beta, \gamma, \tau) \in (0,1) \times [0,1]^2$ be the effective parameters of Section 3.1. Let $\Delta := f(x_0) - f_{\inf}$ and initialize m_0 by averaging $B_0 = \lceil 1/\beta \rceil$ independent stochastic gradients at x_0. In the variance-reduction regime $0 < \varepsilon \lesssim \chi_\tau \sigma_*$, choose $\beta \asymp \varepsilon^2/(\chi_\tau^2 \sigma_*^2)$ and $\eta \asymp \min\{\varepsilon/L, \varepsilon\sqrt{\beta}/\delta_{\gamma,*}, \beta\varepsilon/((1-\gamma)L)\}$, where terms with zero denominator are omitted. Then VENUS returns $\widehat{x} := x_R$ satisfying $\mathbb{E}\big[\|\nabla f(\widehat{x})\|_*\big] \le O(\varepsilon)$ after*

$$N = O\left(\frac{\chi_\tau \sigma_*^2 + L\Delta}{\varepsilon^2} + \frac{\chi_\tau \delta_{\gamma,*} \sigma_* \Delta}{\varepsilon^3} + \frac{\chi_\tau^2 (1-\gamma) L \sigma_*^2 \Delta}{\varepsilon^4} \right) \tag{12}$$

iterations, corresponding to $B_0 + 3N$ stochastic-gradient evaluations. The finite-horizon result and proof are in Appendix B.

Comparison with coupled MARS. At fixed effective (β, γ), $\tau = 0$ recovers coupled MARS, whereas $\tau = 0.5$ gives $\chi_\tau = 1/\sqrt{2}$. In the common iteration bound, the explicit coefficients of the noise-dependent σ_*^2/ε^2 term and the ε^{-3} term decrease by a factor of $1/\sqrt{2}$, and that of the ε^{-4} term by a factor of $1/2$; the $L\Delta/\varepsilon^2$ contribution is unchanged. Accounting for three equal-cost minibatch-gradient evaluations for VENUS and two for MARS gives multipliers of $3/(2\sqrt{2}) \approx 1.06$ for the first two terms, $3/4$ for the ε^{-4} term and $3/2$ for $L\Delta/\varepsilon^2$. Hence the LMO bound certifies an oracle-cost improvement only in regimes where the $(1-\gamma)$-dependent drift term is sufficiently important; it does not establish uniform dominance. This call-count comparison assumes equal-cost calls to the same stochastic oracle, whereas the experiments separately match per-example gradient work. For the ordinary Euclidean update $x_{t+1} = x_t - \eta m_t$, Appendix B.5 gives a matched proof-bound comparison in which both high-order coefficients have a ratio of $3/4$, while the shared $L\Delta/\varepsilon^2$ coefficient has a ratio of $3/2$.

4 CIFAR-10 CNN STUDY AT EQUAL COMPUTE

On CIFAR-10 we compare VENUS against MARS, its coupled special case, and AdamW. VENUS and MARS share the same Scion/Gluon LMO geometry and differ *only* in the momentum estimator; AdamW is the Euclidean baseline. Each optimizer is tuned independently and we report multi-seed means $\pm$ SEM. The architecture, held-out validation protocol, tuning pipeline, best configurations, and full multi-regime study are deferred to Appendix C.

Large batch, equal gradients per step. VENUS requires three stochastic-gradient evaluations per step, MARS two and AdamW one, so an equal-*iteration* comparison would not be an equal-*compute* one. We separate two units: a *data pass* is one traversal of the $N = 45{,}000$ training images, and an *epoch* is N per-example gradient computations, so a data pass of VENUS or MARS costs three or two epochs. At a fixed number of data passes the total work is passes $\times\, N \times$ (grads per example) and *the batch size cancels*, so the compute budget must be matched by adjusting the number of data passes. We match all three at 3000 gradients per step: VENUS at batch 1000 for 32 data passes, MARS at 1500 for 48, AdamW at 3000 for 96. All three use the same nominal budget of 96 epochs (4.32M per-example gradients) and 1440 optimizer steps. Evaluation schedules and checkpoint-selection procedures are detailed in Appendix C. All use a constant stepsize and label-smoothed cross-entropy. **Outcome:** In this regime VENUS achieves the best values on all three metrics (Table 1), with margins over MARS, the stronger baseline, that substantially exceed the corresponding standard errors (0.0003–0.0005). Additional compute provides only a limited improvement for MARS in this regime: increasing its budget from 2.88M to 4.32M changes its validation accuracy only from 0.9261 to 0.9273. AdamW is the weakest of the three at this batch size. **Other regimes:** Across the tested regimes, the selected VENUS configurations have favourable mean summary metrics relative to MARS, although the learning curves do not show uniform dominance and the test-accuracy difference at batch size 128 remains unresolved. At equal batch sizes, AdamW generally achieves better single-view metrics in late training.

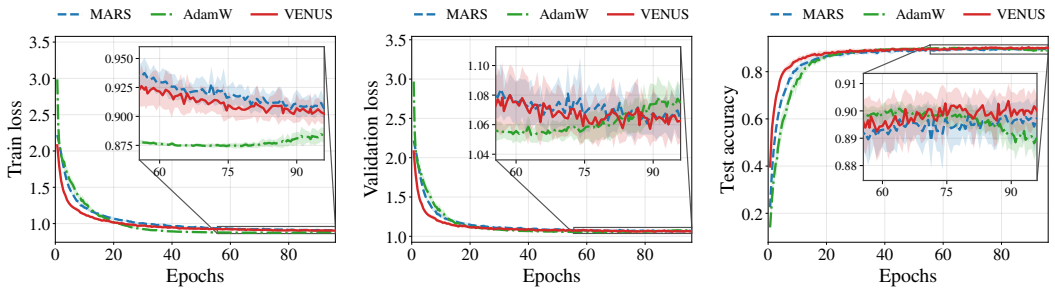


Figure 1: Learning curves in the equal-gradients-per-step regime, against epochs of gradient work (mean over 20 seeds; bands are ± one standard deviation; insets zoom the converged tail). Left to right: train loss, validation loss, test accuracy.

Metric	VENUS (decoupled)	MARS (coupled)	AdamW ($\lambda = 0$)
Validation accuracy ↑	$\mathbf{0.9310 \pm 0.0003}$	0.9273 ± 0.0003	0.9240 ± 0.0004
Validation loss ↓	$\mathbf{1.0376 \pm 0.0006}$	1.0464 ± 0.0005	1.0461 ± 0.0010
Test acc. at best val ↑	$\mathbf{0.9241 \pm 0.0005}$	0.9194 ± 0.0004	0.9156 ± 0.0004

Table 1: CIFAR-10 CNN, large batch at **equal gradients per step**. Mean ± SEM (VENUS/MARS: 50 seeds; AdamW: 20); best in bold.

5 Language-model pretraining at matched gradient budgets

Protocol. We pretrain a 124M-parameter Transformer on FineWeb at branch batch sizes 16, 32 and 64, comparing exact MARS and VENUS estimators with a common spectral-norm LMO update. Each run receives a budget of 2.70B gradient tokens, counting tokens once per gradient evaluation. Exact MARS requires two gradient evaluations per step and VENUS three, so VENUS performs fewer updates. Hyperparameters are selected independently for each method and batch size on one tuning seed, followed by confirmation on six additional paired seeds. This budget matches gradient work, but not data exposure or wall-clock time. The full protocol and tuning budgets are reported in Appendix D.

Choice of the sample-splitting weight. At batch size 32, $\tau = 0.5$ achieves lower final validation loss than $\tau = 0.1$ and $\tau = 0.75$ in all 32 common configurations on the tuning seed, and outperforms MARS in 31 of them (Figure 2). This agrees with the minimum of the theoretical factor $\chi_\tau = \sqrt{\tau^2 + (1-\tau)^2}$, attained at $\tau = 0.5$ (Section 3.2.2). We therefore fix $\tau = 0.5$ in the batch-16 and batch-64 studies. Full results are reported in Appendix D.4.2; the finer CNN sensitivity study is in Appendix C.7.

Sensitivity to the stepsize. At batch size 32, VENUS shows less variation in final validation loss across the shared tested stepsizes in two of the three examined β settings (Figure 3). This observation is specific to the tuning seed and does not establish uniformly lower sensitivity. Full numerical comparisons are reported in Appendix D.4.4.

Batch-dependent performance of the exact estimators. The validation-loss curves and their paired differences (Figure 4) show that the benefit of sample decoupling depends on both batch size and training stage. At batch sizes 16 and 32, VENUS has lower mean validation loss at every scheduled checkpoint, although the gap narrows late in training. At the final checkpoint, the paired confidence interval supports an advantage at batch size 16, while the difference at batch size 32 remains inconclusive (Table 2). At batch size 64, VENUS has a temporary advantage at intermediate checkpoints, but MARS achieves lower final validation loss. Across the tested configurations, the reduction in final validation loss associated with increasing batch size exceeds the difference between estimators at a fixed batch size. Full results are reported in Appendix D.4.

Cached-gradient variants. Following the cached-gradient approximations used in MARS and MARS-M (Yuan et al., 2025; Liu et al., 2025b), we also evaluate cached versions of our MARS and

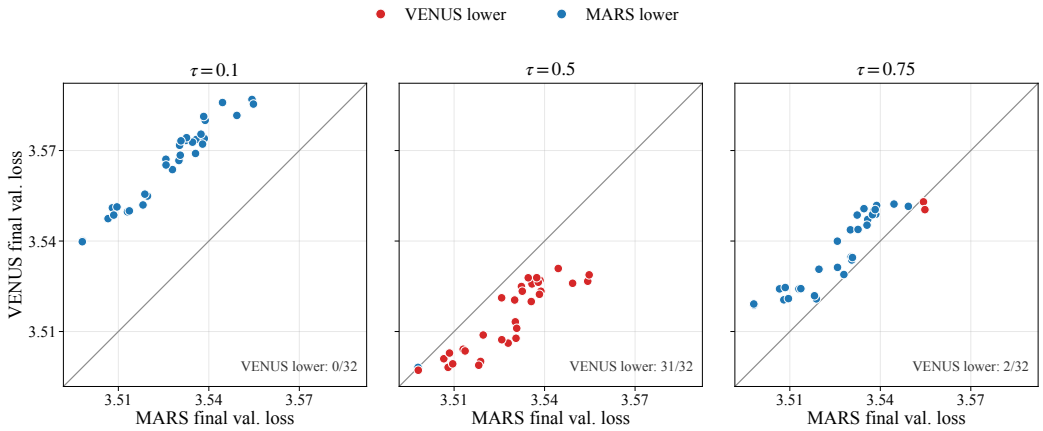


Figure 2: Final validation loss of VENUS versus MARS at batch size 32 on the tuning seed. Each panel contains the same 32 completed (η, β, γ) configurations. Points below the diagonal favour VENUS. Extended comparisons are reported in Appendix D.4.2.

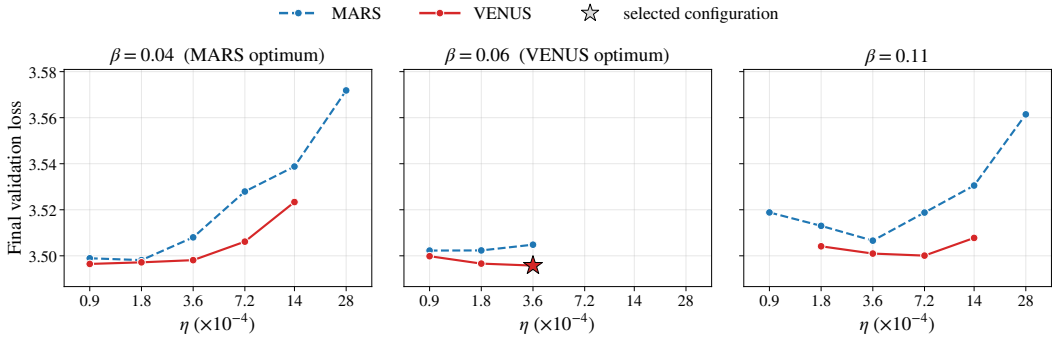


Figure 3: Batch size 32, tuning seed: final validation loss versus peak stepsize η at three values of β, with $\gamma = 0.0015$ and $\tau = 0.5$ for VENUS. The loss-range comparison uses only stepsizes tested for both methods; MARS has additional evaluations at larger stepsizes.

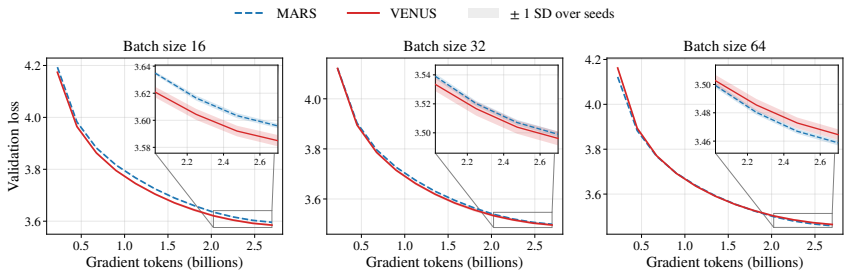
VENUS estimators. Reusing the previous-step gradient reduces their cost to one and two gradient evaluations per step, respectively, allowing more updates under the same gradient-token budget. At batch sizes 16 and 32, cached MARS achieves lower final validation loss than cached VENUS, and both outperform the exact variants on all six confirmation seeds. Figure 5 shows that cached VENUS remains between cached MARS and the exact methods at every scheduled validation checkpoint; this ordering holds on each confirmation seed.

These variants fall outside our exact-transport analysis. Caching changes the estimator, update count and sampled-data exposure simultaneously, so the comparison does not isolate their individual effects. The cached methods also use unequal tuning grids, with $\tau = 0.5$ fixed for cached VENUS. Thus the results characterize the tested configurations rather than establish a general ordering of cached estimators. Definitions, tuning budgets and full results are reported in Appendix D.5.

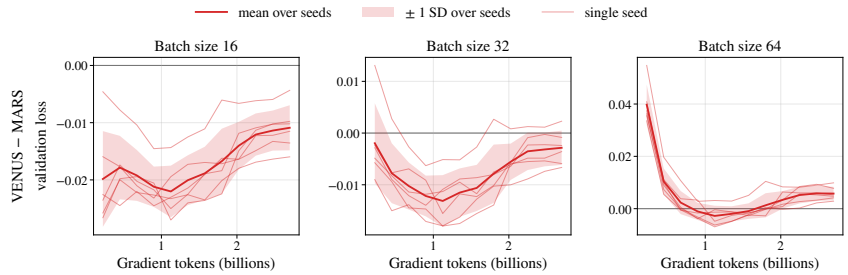
6 Estimator variance and optimization progress

The preceding experiments compare complete training runs. We now examine how decoupling changes the noise in a single estimator update, and whether that change improves optimization after accounting for its computational cost.

Noise at a fixed training state. Hold (x_k, x_{k+1}, m_k) and (β, γ) fixed and let $\mathbb{E}_k$ average over fresh, independent minibatches ξ and ζ of equal size and distribution. Using g^+, g^0, h^+ from Algorithm 2,



(a) Validation loss.



(b) Paired validation-loss differences, VENUS minus MARS.

Figure 4: Selected exact configurations at batch sizes 16, 32 and 64, evaluated at the 12 scheduled checkpoints against gradient tokens. Thick curves show means over six confirmation seeds; bands show ± 1 sample standard deviation, not confidence intervals. In (a), insets magnify the final quarter of the budget. In (b), thin lines show individual paired differences; negative values favour VENUS, and the vertical scales differ between panels. Final results and paired confidence intervals are reported in Table 2.

Batch size	MARS	VENUS	Mean difference	95% confidence interval
16	3.5958	3.5849	−0.0109	[−0.0150, − 0.0068]
32	3.4990	3.4961	−0.0028	[−0.0063, + 0.0006]
64	3.4588	3.4646	+0.0058	[+0.0031, + 0.0085]

Table 2: Final validation loss averaged over six confirmation seeds. The paired difference is VENUS minus MARS; negative values favour VENUS. Intervals are unadjusted 95% paired Student-t confidence intervals for the mean difference.

with iteration subscripts omitted, define the centered fresh-gradient noises $z^\zeta = g^+ - \mathbb{E}_k g^+$ and $z^\xi = h^+ - \mathbb{E}_k h^+$, and the centered transport noise $\widetilde{r}^\zeta = \gamma[(g^+ - g^0) - \mathbb{E}_k(g^+ - g^0)]$. The centered innovations are $q^{\mathrm{MARS}} = (1-\beta)\widetilde{r}^\zeta + \beta z^\zeta$ and $q^{\mathrm{VENUS}}(\tau) = (1-\beta)\widetilde{r}^\zeta + \beta(1-\tau)z^\zeta + \beta\tau z^\xi$. Each vector equals $m_{k+1} - \mathbb{E}_k m_{k+1}$. Its second moment, $\mathbb{E}_k\|q\|_2^2$, is the *innovation variance*: noise from the new minibatches, measured before the LMO in Euclidean norm. The mean-squared error relative to $\nabla f(x_{k+1})$ also includes squared conditional bias from the incoming estimate and imperfect transport.

Two contributions to the variance difference. Write $V = \mathbb{E}_k\|z^\zeta\|_2^2$ and $C = \mathbb{E}_k\langle \widetilde{r}^\zeta, z^\zeta\rangle$. Here V measures fresh-gradient noise, while C measures whether transport and fresh-gradient noise reinforce one another ($C > 0$) or partially cancel ($C < 0$). Independence of z^ξ from the entire ζ contribution gives

$$D(\tau) := \mathbb{E}_k\|q^{\mathrm{MARS}}\|_2^2 - \mathbb{E}_k\|q^{\mathrm{VENUS}}(\tau)\|_2^2 = 2\beta^2\tau(1-\tau)V + 2\beta(1-\beta)\tau C. \tag{13}$$

The conditional means agree, so D also equals the mean-squared-error difference at this fixed state, not across different training trajectories. Positive D favors VENUS. The first term comes from averaging independent fresh gradients and is largest at $\tau = 0.5$. The second reflects reduced same-sample covariance: removing positive covariance helps, whereas removing negative covariance

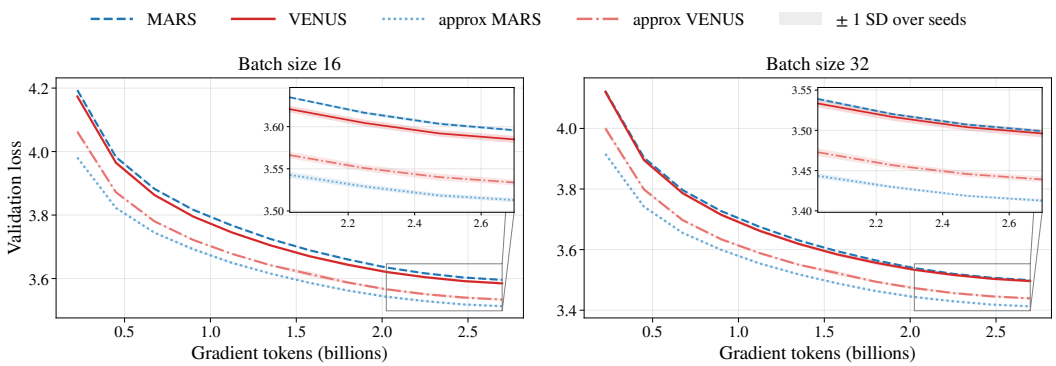


Figure 5: Validation loss of the selected exact and cached configurations at batch sizes 16 and 32, against gradient tokens. Curves show means over six confirmation seeds at the 12 scheduled checkpoints; bands are ± 1 sample standard deviation, not confidence intervals. The legend's "approx" denotes the cached variants. Insets magnify the final quarter of the budget.

sacrifices cancellation. For $V > 0$, minimizing the actual variance at this state gives $\tau^*_{\mathrm{var}} = \Pi_{[0,1]}\big(0.5+(1-\beta)C/(2\beta V)\big)$, where $\Pi_{[0,1]}$ clips its argument to $[0,1]$. Thus the variance-minimizing split is 0.5 when $C = 0$, shifts upward for positive covariance, and downward for negative covariance. The convergence bound uses $\chi_\tau = \sqrt{\tau^2 + (1-\tau)^2}$, which is minimized at 0.5 independently of C. This factor does not retain the measured covariance of each state, whereas the exact variance does. Appendix E constructs examples where covariance reverses the variance ordering. In the CIFAR-10 covariance study, negative covariance occurs but is too weak to reverse $D(0.5) > 0$ in the measured states (Appendix F). These CIFAR-10 diagnostic studies use $\gamma = 1$. Reducing γ at fixed states attenuates the covariance contribution, but this calculation does not predict the states visited by a separately trained model.

From variance to optimization progress. Appendix G holds the model and incoming estimator fixed at training states reconstructed by replay, then draws repeated minibatches to compare updates. Replay closely reproduces loss, but covariance changes sign in seven of 112 states; all branches use the same replayed state. Progress is measured on the finite-oracle training objective, the average loss over a fixed collection of augmented minibatches. At equal branch batch size, larger D is associated with better one-step Euclidean progress, but VENUS spends three gradient evaluations against two for MARS. Giving MARS a batch of $1.5b$ matches the $3b$ per-example gradient work of VENUS at batch b. At $\tau = 0.5$, the Euclidean update favors MARS on average, whereas the layerwise LMO favors VENUS, without establishing a lower mean estimator squared error. Thus Euclidean variance alone does not determine the LMO benefit. The layerwise LMO comparison against the higher-work MARS baseline at $2b$ remains inconclusive. In the separate equal-work comparison of three Euclidean MARS updates with two VENUS updates at $\tau = 0.5$, VENUS is favored in only 13 of 48 tested states, while the mean difference remains inconclusive. These local tests do not establish full-training outcomes.

Diagnostics during Transformer training. At batch size 16 with $\gamma = 0.0015$, we estimate per-coordinate versions of V, C and D along three GPT-2/FineWeb trajectories using the gradients already computed by VENUS (Appendix H). These observations are time-correlated, not independent experimental repetitions. In the phase averages, the covariance contribution accounts for approximately 1.6% of the estimated variance difference early in training and is negative but below 0.06% in magnitude in the middle and final thirds. Independent-refresh averaging therefore dominates the measured innovation-variance reduction in these runs. Phase-level estimates of τ^*_{var} remain close to 0.5. These measurements support the balanced split's local noise interpretation; they do not establish that variance reduction causes the validation-loss improvement or predicts the batch-size crossover in Section 5.

AI USE STATEMENT

Generative AI tools were used primarily to revise language and grammar and as coding assistants for experimental and analysis scripts. They also assisted with translation, feedback on experimental design, and analysis and interpretation of experimental results. Additional uses included literature search, figure preparation and manuscript organization. The authors are responsible for verifying AI-assisted outputs and for the final text, code, results and claims.

REPRODUCIBILITY STATEMENT

Algorithms 1 and 2 specify the exact estimators, and Appendix D.5 defines the cached variants. Assumptions and proofs are provided in Appendices A and B. The CNN and language-model protocols, including architectures, datasets, hyperparameter searches, evaluation procedures, seeds and computational budgets, are documented in Appendices C and D. Appendices E, F and G describe the controlled constructions and diagnostic experiments, and Appendix H reports the Transformer covariance diagnostics. The experimental appendices also report the detailed results and statistical summaries supporting the empirical comparisons.

REFERENCES

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

Åke Björck and Clazett Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Sébastien Bubeck, Christian Coester, Ronen Eldan, Timothy Gowers, Yin Tat Lee, Alexandru Lupsasca, Mehtaab Sawhney, Robert Scherrer, Mark Sellke, Brian K Spears, et al. Early science acceleration experiments with GPT-5. *arXiv preprint arXiv:2511.16072*, 2025.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kaja Gruntkowska, Yassine Maziane, Zheng Qu, and Peter Richtárik. Drop-Muon: Update less, converge faster. *arXiv preprint arXiv:2510.02239*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

Keller Jordan. 94% on cifar-10 in 3.29 seconds on a single gpu. *arXiv preprint arXiv:2404.00498*, 2024.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, Vlado Boza, You Jiacheng, Franz Cesista, and Braden Koszarsky. modded-nanogpt: Speedrunning the NanoGPT baseline. `https://github.com/KellerJordan/modded-[]nanogpt`, 2024a.

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

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *The 3rd International Conference on Learning Representations*, 2015. URL `https://arxiv.org/pdf/1412.6980.pdf`.

Deyi Liu, Lam M Nguyen, and Quoc Tran-Dinh. An optimal hybrid variance-reduced algorithm for stochastic composite nonconvex optimization. *arXiv preprint arXiv:2008.09055*, 2020.

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.

Yifeng Liu, Angela Yuan, and Quanquan Gu. MARS-M: When variance reduction meets matrices. *arXiv preprint arXiv:2510.21800*, 2025b. v3, 29 January 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Antonio Orvieto and Robert Gower. In search of Adam's secret sauce. *Advances in Neural Information Processing Systems*, 38:63404–63442, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained lmos. *arXiv preprint arXiv:2502.07529*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL `https://cdn.openai.com/better-[]language-[]models/language_models_are_unsupervised_multitask_learners.pdf`.

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making Muon & Scion great again!(bridging theory and practice of LMO-based optimizers for LLMs). *arXiv preprint arXiv:2505.13416*, 2025.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. *arXiv preprint arXiv:2509.01440*, 2025.

Egor Shulgin, Tamaz Gadaev, Sarit Khirirat, and Peter Richtárik. Understanding MARS: When scaling momentum provably helps. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2026. URL https://openreview.net/forum?id=jokamk4yGt.

David So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V Le. Searching for efficient transformers for language modeling. *Advances in neural information processing systems*, 34:6010–6022, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Quoc Tran-Dinh, Nhan H Pham, Dzung T Phan, and Lam M Nguyen. Hybrid stochastic gradient descent algorithms for stochastic nonconvex optimization. *arXiv preprint arXiv:1905.05920*, 2019.

Quoc Tran-Dinh, Nhan H Pham, Dzung T Phan, and Lam M Nguyen. A hybrid stochastic optimization framework for composite nonconvex optimization. *Mathematical Programming*, 191(2):1005–1071, 2022.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 93423–93444, 2025.

Huizhuo Yuan, Yifeng Liu, Shuang Wu, Xun Zhou, and Quanquan Gu. MARS: Unleashing the power of variance reduction for training large models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025. arXiv:2411.10438v4.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

A Deferred material for Sections 2 and 3.2.1

A.1 Discussion of the assumptions

Condition (A) is the γ-similarity condition: it measures how well the scaled stochastic transport $\gamma d_\xi(x,y)$ tracks the population difference $d(x,y)$. It reduces to standard mean-squared similarity at $\gamma = 1$ and follows from L-smoothness at $\gamma = 0$. More precisely, if

$$\mathbb{E}_\xi \|d_\xi(x,y) - d(x,y)\|_*^2 \le \delta^2 \|x - y\|^2,$$

then, for $\gamma \in [0,1]$, one may take $\delta_\gamma \le \gamma\delta + (1-\gamma)L$. In Euclidean geometry, unbiasedness gives the sharper admissible choice

$$\delta_\gamma^2 \le \gamma^2\delta^2 + (1-\gamma)^2 L^2.$$

Thus γ-similarity imposes no stronger regularity than standard similarity and smoothness. A smaller δ_γ is accompanied, however, by the explicit $(1-\gamma)$-dependent drift term in the complexity bound.

A.2 Proof of the shared-sample MARS decomposition

Proof of Proposition 1. For compactness, denote the two stochastic gradients evaluated on the shared minibatch by

$$g_k^+ := \nabla f_{\xi_k}(x_k), \qquad g_k^0 := \nabla f_{\xi_k}(x_{k-1}),$$

and abbreviate the convex combination in (8) by

$$\overline{m}_k := \mathrm{Conv}_\theta\big(m_k^{\mathrm{MSGD}}(\mu), m_k^{\mathrm{MVR}}(p)\big) = (1-\theta)m_k^{\mathrm{MSGD}}(\mu) + \theta m_k^{\mathrm{MVR}}(p).$$

The two candidates have the same incoming estimate m_{k-1}. Substituting their definitions from (3) and (4), and first simplifying the MVR candidate, gives

$$\begin{aligned} m_k^{\mathrm{MSGD}}(\mu) &= (1-\mu)m_{k-1} + \mu g_k^+, \\ m_k^{\mathrm{MVR}}(p) &= (1-p)\big(m_{k-1} + g_k^+ - g_k^0\big) + p g_k^+ \\ &= (1-p)m_{k-1} + \big((1-p)+p\big)g_k^+ - (1-p)g_k^0 \\ &= (1-p)m_{k-1} + g_k^+ - (1-p)g_k^0. \end{aligned}$$

We now expand the convex combination without suppressing any terms:

$$\begin{aligned} \overline{m}_k &= (1-\theta)\big((1-\mu)m_{k-1} + \mu g_k^+\big) \\ &\quad + \theta\big((1-p)m_{k-1} + g_k^+ - (1-p)g_k^0\big) \\ &= (1-\theta)(1-\mu)m_{k-1} + (1-\theta)\mu g_k^+ \\ &\quad + \theta(1-p)m_{k-1} + \theta g_k^+ - \theta(1-p)g_k^0 \\ &= \big((1-\theta)(1-\mu) + \theta(1-p)\big)m_{k-1} \\ &\quad + \big((1-\theta)\mu + \theta\big)g_k^+ - \theta(1-p)g_k^0. \end{aligned}$$

It remains to identify these three coefficients with those of a MARS update. By the definition $\beta = (1-\theta)\mu + \theta p$, the coefficient of the incoming estimate is

$$\begin{aligned} (1-\theta)(1-\mu) + \theta(1-p) &= (1-\theta) - (1-\theta)\mu + \theta - \theta p \\ &= 1 - \big((1-\theta)\mu + \theta p\big) \\ &= 1 - \beta. \end{aligned}$$

The coefficient of the current stochastic gradient can be split into its MARS fresh-gradient part and correction part as

$$\begin{aligned} (1-\theta)\mu + \theta &= (1-\theta)\mu + \theta p + \theta(1-p) \\ &= \beta + \theta(1-p). \end{aligned}$$

Substituting these two identities into the expanded mixture yields

$$\begin{aligned} \overline{m}_k &= (1-\beta)m_{k-1} + \big(\beta + \theta(1-p)\big)g_k^+ - \theta(1-p)g_k^0 \\ &= \beta g_k^+ + (1-\beta)m_{k-1} + \theta(1-p)(g_k^+ - g_k^0). \end{aligned}$$

The effective correction scale in (7) is chosen precisely so that $\gamma = \theta(1-p)/(1-\beta)$, or equivalently $\theta(1-p) = (1-\beta)\gamma$. Therefore, substituting this identity into the last display proves (8):

$$\begin{aligned}\overline{m}_k &= \beta g_k^+ + (1-\beta)m_{k-1} + (1-\beta)\gamma(g_k^+ - g_k^0) \\ &= (1-\beta)\big(m_{k-1} + \gamma(g_k^+ - g_k^0)\big) + \beta g_k^+ \\ &= m_k^{\mathrm{MARS}}(\beta,\gamma).\end{aligned}$$

It remains to check that the effective correction scale lies in $[0,1]$. By assumption $\beta < 1$, so $1-\beta > 0$ and γ is well defined and nonnegative. Moreover

$$1-\beta = (1-\theta)(1-\mu) + \theta(1-p) \geq \theta(1-p),$$

and dividing this inequality by $1-\beta > 0$ gives $\gamma \leq 1$.

Finally, consider the converse direction. Given any $m_k^{\mathrm{MARS}}(\beta,\gamma)$ with $\beta \in (0,1)$ and $\gamma \in [0,1]$, set $\mu = p = \beta$ and $\theta = \gamma$. The corresponding convex combination expands as

$$\begin{aligned}&(1-\gamma)m_k^{\mathrm{MSGD}}(\beta) + \gamma m_k^{\mathrm{MVR}}(\beta) \\ &= (1-\gamma)\big((1-\beta)m_{k-1} + \beta g_k^+\big) \\ &\quad + \gamma\big((1-\beta)m_{k-1} + g_k^+ - (1-\beta)g_k^0\big) \\ &= (1-\beta)m_{k-1} + \big((1-\gamma)\beta + \gamma\big)g_k^+ - \gamma(1-\beta)g_k^0 \\ &= (1-\beta)m_{k-1} + \big(\beta + \gamma(1-\beta)\big)g_k^+ - \gamma(1-\beta)g_k^0 \\ &= (1-\beta)\big(m_{k-1} + \gamma(g_k^+ - g_k^0)\big) + \beta g_k^+ \\ &= m_k^{\mathrm{MARS}}(\beta,\gamma).\end{aligned}$$

Equivalently, the effective-parameter map is recovered exactly because

$$\begin{aligned}(1-\theta)\mu + \theta p &= (1-\gamma)\beta + \gamma\beta = \beta, \\ \frac{\theta(1-p)}{1-\beta} &= \frac{\gamma(1-\beta)}{1-\beta} = \gamma.\end{aligned}$$

All identities hold pointwise for every realization of the shared minibatch ξ_k; no expectation argument is required. □

B PROOFS FOR VENUS

B.1 ASSUMPTIONS AND EFFECTIVE PARAMETRIZATION

Let $\|\cdot\|$ be the norm defining the LMO ball and let $\|\cdot\|_*$ be its dual. Throughout this appendix, we assume that $f_{\inf} := \inf_x f(x) > -\infty$ and that

$$\|\nabla f(x) - \nabla f(y)\|_* \le L\|x-y\|, \qquad x,y \in \mathbb{R}^d. \tag{14}$$

Consequently,

$$f(y) \le f(x) + \langle \nabla f(x), y-x\rangle + \frac{L}{2}\|y-x\|^2. \tag{15}$$

The stochastic gradients satisfy, for every x,

$$\mathbb{E}_\xi[\nabla f_\xi(x)] = \nabla f(x), \qquad \mathbb{E}_\xi\|\nabla f_\xi(x) - \nabla f(x)\|_2^2 \le \sigma^2. \tag{16}$$

At every iteration, ξ_{k+1} and ζ_{k+1} are fresh, conditionally independent, and distributed as the stochastic oracle in (16). We also assume that there are constants $0 < \rho \le \bar\rho < \infty$ such that

$$\rho\|v\|_2 \le \|v\|_* \le \bar\rho\|v\|_2, \qquad v \in \mathbb{R}^d. \tag{17}$$

For the fixed effective correction scale $\gamma \in [0,1]$, suppose that there is a constant $\delta_\gamma \ge 0$ satisfying

$$\mathbb{E}_\zeta \left\|\gamma\big(\nabla f_\zeta(x) - \nabla f_\zeta(y)\big) - \big(\nabla f(x) - \nabla f(y)\big)\right\|_*^2 \le \delta_\gamma^2\|x-y\|^2 \tag{18}$$

for all x,y. Define the geometry-adjusted constants

$$\sigma_* := \bar\rho\sigma, \qquad \delta_{\gamma,*} := \frac{\bar\rho}{\rho}\delta_\gamma. \tag{19}$$

For the Euclidean LMO, $\rho = \bar\rho = 1$.

We next verify the effective form used in the main text. Recall that θ is the outer mixing weight in Algorithm 2. Expanding the two branches gives

$$\begin{aligned} m_{k+1} = {} & \big[(1-\theta)(1-\mu) + \theta(1-p)\big]m_k + (1-\theta)\mu\nabla f_{\xi_{k+1}}(x_{k+1}) \\ & + \theta p\nabla f_{\zeta_{k+1}}(x_{k+1}) + \theta(1-p)\big(\nabla f_{\zeta_{k+1}}(x_{k+1}) - \nabla f_{\zeta_{k+1}}(x_k)\big). \end{aligned} \tag{20}$$

Let

$$\beta := (1-\theta)\mu + \theta p, \qquad \tau := \frac{(1-\theta)\mu}{\beta}, \qquad \gamma := \frac{\theta(1-p)}{1-\beta}, \tag{21}$$

where $0 < \beta < 1$. Then

$$(1-\theta)\mu = \beta\tau, \qquad \theta p = \beta(1-\tau), \qquad \theta(1-p) = (1-\beta)\gamma, \tag{22}$$

and the coefficient of m_k in (20) is $1-\beta$. Hence

$$\begin{aligned} m_{k+1} = {} & (1-\beta)\Big(m_k + \gamma\big[\nabla f_{\zeta_{k+1}}(x_{k+1}) - \nabla f_{\zeta_{k+1}}(x_k)\big]\Big) \\ & + \beta\Big[\tau\nabla f_{\xi_{k+1}}(x_{k+1}) + (1-\tau)\nabla f_{\zeta_{k+1}}(x_{k+1})\Big]. \end{aligned} \tag{23}$$

Conversely, every $(\beta,\gamma,\tau) \in (0,1)\times[0,1]^2$ is represented in the continuous closure of the primitive parametrization by

$$\theta = \beta(1-\tau) + (1-\beta)\gamma, \qquad p = \frac{\beta(1-\tau)}{\theta}, \qquad \mu = \frac{\beta\tau}{1-\theta}, \tag{24}$$

with the zero-denominator cases interpreted as follows: if $\theta = 0$, the MVR branch is absent and p is arbitrary; if $\theta = 1$, the MSGD branch is absent and μ is arbitrary. Indeed,

$$1-\theta = \beta\tau + (1-\beta)(1-\gamma),$$

which shows that the nontrivial fractions in (24) lie in $[0,1]$. Boundary values such as $\tau = 0$ may require $\mu = 0$ and are therefore understood through this closure, or directly through the effective update (23). In particular, $\tau = 0$ is precisely the coupled-MARS estimator, with the unused ξ-gradient omitted.

B.2 LMO DESCENT AND ESTIMATOR TRACKING

For a constant radius $\eta > 0$, let

$$s_k \in \arg\min_{\|s\| \le \eta} \langle m_k, s \rangle, \qquad x_{k+1} = x_k + s_k, \qquad e_k := m_k - \nabla f(x_k). \tag{25}$$

Lemma 1 (LMO descent)**.** *For every $k \ge 0$,*

$$\eta \|\nabla f(x_k)\|_* \le f(x_k) - f(x_{k+1}) + 2\eta \|e_k\|_* + \frac{L\eta^2}{2}. \tag{26}$$

Proof. By exactness of the LMO and the definition of the dual norm,

$$\langle m_k, s_k \rangle = -\eta \|m_k\|_*.$$

Since $m_k = \nabla f(x_k) + e_k$ and $\|s_k\| \le \eta$,

$$\begin{aligned} \langle \nabla f(x_k), s_k \rangle &= \langle m_k, s_k \rangle - \langle e_k, s_k \rangle \\ &\le -\eta \|m_k\|_* + \eta \|e_k\|_* \\ &\le -\eta \|\nabla f(x_k)\|_* + 2\eta \|e_k\|_*. \end{aligned}$$

Applying (15) with $y = x_{k+1} = x_k + s_k$ proves the claim. □

Define

$$\Psi_\tau := \Big(\beta\tau\sigma_*)^2 + [(1-\beta)\delta_{\gamma,*}\eta + \beta(1-\tau)\sigma_*]^2 \Big)^{1/2}. \tag{27}$$

This quantity is a covariance-robust upper bound on the centered innovation generated in one estimator update.

Lemma 2 (Tracking error)**.** *Let $a := 1 - \beta$. For every $k \ge 0$,*

$$\mathbb{E}\|e_k\|_* \le a^k \mathbb{E}\|e_0\|_* + \frac{a(1-\gamma)L\eta}{\beta}(1 - a^k) + \Psi_\tau \sqrt{\frac{1 - a^{2k}}{\beta(2-\beta)}}. \tag{28}$$

Consequently, if $N = K + 1$, then

$$\sum_{k=0}^{K} \mathbb{E}\|e_k\|_* \le \frac{\mathbb{E}\|e_0\|_*}{\beta} + N\frac{(1-\beta)(1-\gamma)L\eta}{\beta} + N\frac{\Psi_\tau}{\sqrt{\beta(2-\beta)}}. \tag{29}$$

Proof. Let

$$\mathcal{F}_k := \sigma(x_0, m_0, \xi_1, \zeta_1, \ldots, \xi_k, \zeta_k), \qquad \mathbb{E}_k[\cdot] := \mathbb{E}[\cdot \mid \mathcal{F}_k].$$

Then x_k, x_{k+1} and s_k are $\mathcal{F}_k$-measurable, whereas ξ_{k+1} and ζ_{k+1} are fresh and conditionally independent. Write

$$F_k := \nabla f(x_k), \qquad D_{k+1} := F_{k+1} - F_k,$$

and define

$$\begin{aligned} z^{\xi}_{k+1} &:= \nabla f_{\xi_{k+1}}(x_{k+1}) - F_{k+1}, \\ z^{\zeta}_{k+1} &:= \nabla f_{\zeta_{k+1}}(x_{k+1}) - F_{k+1}, \\ r^{\zeta}_{k+1} &:= \gamma\big[\nabla f_{\zeta_{k+1}}(x_{k+1}) - \nabla f_{\zeta_{k+1}}(x_k)\big] - D_{k+1}. \end{aligned}$$

Subtracting F_{k+1} from (23) gives

$$e_{k+1} = a e_k + a r^{\zeta}_{k+1} + \beta(1-\tau) z^{\zeta}_{k+1} + \beta\tau z^{\xi}_{k+1}. \tag{30}$$

Unbiasedness implies

$$\mathbb{E}_k[r^{\zeta}_{k+1}] = (\gamma - 1) D_{k+1}.$$

Let

$$\widetilde{r}^{\zeta}_{k+1} := r^{\zeta}_{k+1} - \mathbb{E}_k[r^{\zeta}_{k+1}] = \gamma\Big([\nabla f_{\zeta_{k+1}}(x_{k+1}) - \nabla f_{\zeta_{k+1}}(x_k)] - D_{k+1} \Big)$$

and

$$q_{k+1} := a\widetilde{r}^{\zeta}_{k+1} + \beta(1-\tau)z^{\zeta}_{k+1} + \beta\tau z^{\xi}_{k+1}. \tag{31}$$

Then

$$e_{k+1} = ae_k + a(\gamma-1)D_{k+1} + q_{k+1}, \qquad \mathbb{E}_k[q_{k+1}] = 0. \tag{32}$$

Centering cannot increase Euclidean variance. Thus, by (17) and (18),

$$\mathbb{E}_k\|\widetilde{r}^{\zeta}_{k+1}\|_2^2 \le \mathbb{E}_k\|r^{\zeta}_{k+1}\|_2^2 \le \frac{\delta_\gamma^2}{\rho^2}\|s_k\|^2 \le \frac{\delta_\gamma^2\eta^2}{\rho^2}. \tag{33}$$

The ξ-term is conditionally independent of the complete ζ-block in (31), and both blocks are conditionally centered. Therefore,

$$\mathbb{E}_k\|q_{k+1}\|_2^2 = \beta^2\tau^2\mathbb{E}_k\|z^{\xi}_{k+1}\|_2^2 + \mathbb{E}_k\left\|a\widetilde{r}^{\zeta}_{k+1} + \beta(1-\tau)z^{\zeta}_{k+1}\right\|_2^2.$$

Minkowski's inequality in L_2, followed by (16) and (33), yields

$$\begin{aligned}\left(\mathbb{E}_k\|q_{k+1}\|_2^2\right)^{1/2} &\le \sqrt{(\beta\tau\sigma)^2 + \left(\frac{a\delta_\gamma\eta}{\rho} + \beta(1-\tau)\sigma\right)^2} \\ &= \frac{\Psi_\tau}{\bar{\rho}}.\end{aligned} \tag{34}$$

Unrolling (32), for $k \ge 1$, gives

$$e_k = a^k e_0 + a(\gamma-1)\sum_{t=0}^{k-1} a^{k-1-t}D_{t+1} + \sum_{t=0}^{k-1} a^{k-1-t}q_{t+1}. \tag{35}$$

By (14) and $\|s_t\| \le \eta$,

$$\|D_{t+1}\|_* \le L\eta.$$

Moreover, $(q_{t+1})_{t\ge0}$ is a Euclidean martingale-difference sequence. Hence its cross terms vanish, and

$$\mathbb{E}\left\|\sum_{t=0}^{k-1} a^{k-1-t}q_{t+1}\right\|_2^2 = \sum_{t=0}^{k-1} a^{2(k-1-t)}\mathbb{E}\|q_{t+1}\|_2^2 \le \frac{\Psi_\tau^2}{\bar{\rho}^2}\frac{1-a^{2k}}{1-a^2}.$$

Using Jensen's inequality, the upper inequality in (17), and $1-a^2 = \beta(2-\beta)$, we obtain

$$\mathbb{E}\left\|\sum_{t=0}^{k-1} a^{k-1-t}q_{t+1}\right\|_* \le \Psi_\tau\sqrt{\frac{1-a^{2k}}{\beta(2-\beta)}}. \tag{36}$$

Taking norms in (35), using $\sum_{j=0}^{k-1} a^j = (1-a^k)/\beta$, and recalling that $\gamma \in [0,1]$ proves (28). Finally, $\sum_{k=0}^{K} a^k \le 1/\beta$ and the two remaining terms in (28) are bounded by their stationary values, which proves (29). □

Theorem 2 (Finite-horizon convergence)**.** *Let $N = K+1$, let $R \sim \mathrm{Unif}\{0,\dots,K\}$, and set $\Delta := f(x_0) - f_{\inf}$. Under* (14)–(18)*, the iterates of* (25) *and* (23) *satisfy*

$$\mathbb{E}\|\nabla f(x_R)\|_* \le \frac{\Delta}{N\eta} + \frac{2\mathbb{E}\|e_0\|_*}{\beta N} + \frac{2(1-\beta)(1-\gamma)L\eta}{\beta} + \frac{2\Psi_\tau}{\sqrt{\beta(2-\beta)}} + \frac{L\eta}{2}. \tag{37}$$

Proof. Summing (26) over $k = 0,\dots,K$, taking expectations, and using $f(x_{K+1}) \ge f_{\inf}$ gives

$$\eta\sum_{k=0}^{K}\mathbb{E}\|\nabla f(x_k)\|_* \le \Delta + 2\eta\sum_{k=0}^{K}\mathbb{E}\|e_k\|_* + \frac{NL\eta^2}{2}.$$

Substitute (29), divide by $N\eta$, and use

$$\mathbb{E}\|\nabla f(x_R)\|_* = \frac{1}{N}\sum_{k=0}^{K}\mathbb{E}\|\nabla f(x_k)\|_*.$$

□

B.3 COMPLEXITY CONSEQUENCE

Define

$$\chi_\tau := \sqrt{\tau^2 + (1-\tau)^2}. \tag{38}$$

The triangle inequality in $\mathbb{R}^2$ gives

$$\Psi_\tau \le \beta\chi_\tau\sigma_* + (1-\beta)\delta_{\gamma,*}\eta. \tag{39}$$

If m_0 is the average of $B_0 \ge 1/\beta$ independent stochastic gradients at x_0, then

$$\mathbb{E}\|e_0\|_* \le \frac{\sigma_*}{\sqrt{B_0}} \le \sigma_*\sqrt{\beta}. \tag{40}$$

Combining (37), (39), and $\sqrt{\beta(2-\beta)} \ge \sqrt{\beta}$ yields

$$\begin{aligned}\mathbb{E}\|\nabla f(x_R)\|_* \le \frac{\Delta}{N\eta} + \frac{2\sigma_*}{N\sqrt{\beta}} + 2\chi_\tau\sigma_*\sqrt{\beta} + \frac{2\delta_{\gamma,*}\eta}{\sqrt{\beta}} \\ + \frac{2(1-\gamma)L\eta}{\beta} + \frac{L\eta}{2}.\end{aligned} \tag{41}$$

Corollary 1 (Variance-reduction regime)**.** *Let $0 < \varepsilon \le \chi_\tau\sigma_*$. Choose*

$$\beta := \frac{\varepsilon^2}{144\chi_\tau^2\sigma_*^2}, \qquad B_0 := \left\lceil \frac{1}{\beta} \right\rceil, \tag{42}$$

and

$$\eta := \min\left\{ \frac{\varepsilon}{3L}, \frac{\varepsilon\sqrt{\beta}}{12\delta_{\gamma,*}}, \frac{\beta\varepsilon}{12(1-\gamma)L} \right\}, \tag{43}$$

where a term with zero denominator is omitted. If

$$N \ge \max\left\{ 1, \frac{6\Delta}{\varepsilon\eta}, \frac{12\sigma_*}{\varepsilon\sqrt{\beta}} \right\}, \tag{44}$$

then

$$\mathbb{E}\|\nabla f(x_R)\|_* \le \varepsilon.$$

In particular,

$$N = O\left(\frac{\chi_\tau\sigma_*^2 + L\Delta}{\varepsilon^2} + \frac{\chi_\tau\delta_{\gamma,*}\sigma_*\Delta}{\varepsilon^3} + \frac{\chi_\tau^2(1-\gamma)L\sigma_*^2\Delta}{\varepsilon^4} \right). \tag{45}$$

The total number of stochastic-gradient evaluations is $B_0 + 3N$.

Proof. With (42), the third term on the right-hand side of (41) equals $\varepsilon/6$. The three restrictions in (43) make the fourth, fifth, and sixth terms at most $\varepsilon/6$ each. Finally, (44) makes the first two terms at most $\varepsilon/6$ each. This proves the stationarity claim.

Moreover,

$$\frac{12\sigma_*}{\varepsilon\sqrt{\beta}} = \frac{144\chi_\tau\sigma_*^2}{\varepsilon^2},$$

and

$$\begin{aligned}\frac{6\Delta}{\varepsilon\eta} \le \frac{18L\Delta}{\varepsilon^2} + \frac{864\chi_\tau\delta_{\gamma,*}\sigma_*\Delta}{\varepsilon^3} \\ + \frac{10368\chi_\tau^2(1-\gamma)L\sigma_*^2\Delta}{\varepsilon^4}.\end{aligned}$$

Taking ceilings changes only universal constants and proves (45). Finally, $B_0 = O(\chi_\tau^2\sigma_*^2/\varepsilon^2)$, which is absorbed by the first term in the total oracle cost because $\chi_\tau \in [1/\sqrt{2},1]$. □

Remark 1 (All target accuracies)**.** *The restriction $\varepsilon \le \chi_\tau \sigma_*$ is imposed only to simplify the displayed bound. Taking*

$$\beta = \min\left\{\frac{1}{2}, \frac{\varepsilon^2}{144\chi_\tau^2\sigma_*^2}\right\}$$

and the same form of η gives the globally valid bound

$$N = O\Bigg(1 + \frac{\chi_\tau\sigma_*^2 + (L+\delta_{\gamma,*})\Delta}{\varepsilon^2} + \frac{\chi_\tau\delta_{\gamma,*}\sigma_*\Delta}{\varepsilon^3} + \frac{\chi_\tau^2(1-\gamma)L\sigma_*^2\Delta}{\varepsilon^4}\Bigg).$$

B.4 Comparison with coupled MARS

At fixed effective (β, γ), coupled MARS is the endpoint $\tau = 0$ of (23). To compare the robust tracking bounds, let

$$A := (1-\beta)\delta_{\gamma,*}\eta, \qquad B := \beta\sigma_*.$$

Then

$$\Psi_\tau^2 = B^2\tau^2 + [A + B(1-\tau)]^2, \qquad \Psi_0^2 = (A+B)^2,$$

and direct expansion gives

$$\Psi_0^2 - \Psi_\tau^2 = 2B\tau\,[A + B(1-\tau)] \ge 0. \tag{46}$$

Thus decoupling never worsens this covariance-robust upper bound. If $B > 0$, its minimizer over $\tau \in [0,1]$ is

$$\tau_{\mathrm{bd}}^* = \min\left\{1, \frac{1}{2} + \frac{A}{2B}\right\}, \tag{47}$$

with

$$\min_{\tau\in[0,1]} \Psi_\tau = \begin{cases} (A+B)/\sqrt{2}, & A \le B, \\ \sqrt{A^2+B^2}, & A > B. \end{cases} \tag{48}$$

The minimizing value approaches an equal split when fresh-gradient noise dominates and complete sample separation when correction uncertainty dominates.

Exact variance and the covariance caveat. The comparison above concerns a worst-case bound under (16) and (18); it does not claim exact conditional-variance dominance for every stochastic oracle. In Euclidean geometry, define

$$V_{k+1} := \mathbb{E}_k\|z_{k+1}^\zeta\|_2^2, \qquad C_{k+1} := \mathbb{E}_k\langle \widetilde{r}_{k+1}^\zeta, z_{k+1}^\zeta\rangle.$$

At the same state and effective (β, γ), the centered innovations of coupled MARS and VENUS are

$$\begin{aligned} q_{k+1}^{\mathrm{MARS}} &= (1-\beta)\widetilde{r}_{k+1}^\zeta + \beta z_{k+1}^\zeta, \\ q_{k+1}^{\mathrm{VENUS}} &= (1-\beta)\widetilde{r}_{k+1}^\zeta + \beta(1-\tau)z_{k+1}^\zeta + \beta\tau z_{k+1}^\xi. \end{aligned}$$

Since z_{k+1}^ξ is independent of the ζ-block and has the same conditional second moment as z_{k+1}^ζ,

$$\begin{aligned} &\mathbb{E}_k\|q_{k+1}^{\mathrm{MARS}}\|_2^2 - \mathbb{E}_k\|q_{k+1}^{\mathrm{VENUS}}\|_2^2 \\ &\quad = 2\beta\tau\,[\beta(1-\tau)V_{k+1} + (1-\beta)C_{k+1}]. \end{aligned} \tag{49}$$

Therefore exact one-step conditional-variance dominance holds whenever

$$\beta(1-\tau)V_{k+1} + (1-\beta)C_{k+1} \ge 0. \tag{50}$$

A sufficiently negative C_{k+1} means that coupled MARS benefits from same-sample cancellation. No sign condition on C_{k+1} is used in Theorem 2 or Corollary 1.

B.5 Comparison with ordinary Euclidean MARS

This subsection considers the ordinary Euclidean update

$$x_{t+1} = x_t - \eta g_t \tag{51}$$

with estimator

$$g_t = (1-\beta)\big(g_{t-1} + \gamma d_t^{\zeta}\big) + \beta\big[\tau \nabla f_{\xi_t}(x_t) + (1-\tau)\nabla f_{\zeta_t}(x_t)\big], \tag{52}$$

where $d_t^{\zeta} := \nabla f_{\zeta_t}(x_t) - \nabla f_{\zeta_t}(x_{t-1})$ and $\xi_t \perp \zeta_t$. Coupled MARS is the endpoint $\tau = 0$. In this subsection the norm is Euclidean, and we assume

$$\mathbb{E}_{\zeta} \left\| \gamma\big(\nabla f_{\zeta}(x) - \nabla f_{\zeta}(y)\big) - \big(\nabla f(x) - \nabla f(y)\big) \right\|_2^2 \le \delta_\gamma^2 \|x - y\|_2^2. \tag{53}$$

For $\lambda > 1$, define

$$V_{\lambda,\tau} := \tau^2 + \frac{\lambda}{\lambda - 1}(1-\tau)^2 \tag{54}$$

and

$$A_\lambda := \frac{(1-\beta)^3(1-\gamma)^2 L^2}{\beta^2} + \frac{\lambda(1-\beta)^2\delta_\gamma^2}{\beta}. \tag{55}$$

Theorem 3 (Decoupled ordinary Euclidean MARS)**.** *Let $\widehat{x}_T \sim \mathrm{Unif}\{x_0, \dots, x_{T-1}\}$, $e_0 := g_0 - \nabla f(x_0)$, and $\Delta := f(x_0) - f_{\inf}$. If*

$$0 < \eta \le \frac{1}{L + \sqrt{A_\lambda}}, \tag{56}$$

then

$$\mathbb{E}\|\nabla f(\widehat{x}_T)\|_2^2 \le \frac{2\Delta}{\eta T} + \frac{\mathbb{E}\|e_0\|_2^2}{\beta T} + \beta V_{\lambda,\tau}\sigma^2. \tag{57}$$

If g_0 is an average of $B_0 \ge 1/\beta$ independent stochastic gradients, then the middle term is at most σ^2/T.

Proof. Set

$$F_t := \nabla f(x_t), \qquad D_t := F_t - F_{t-1}, \qquad e_t := g_t - F_t, \qquad a := 1 - \beta,$$

and define

$$z_t^{\xi} := \nabla f_{\xi_t}(x_t) - F_t, \qquad z_t^{\zeta} := \nabla f_{\zeta_t}(x_t) - F_t, \qquad r_t^{\zeta} := \gamma d_t^{\zeta} - D_t.$$

Then

$$e_t = a e_{t-1} + w_t, \qquad w_t := a r_t^{\zeta} + \beta(1-\tau) z_t^{\zeta} + \beta\tau z_t^{\xi}. \tag{58}$$

Let $\mathcal{G}_{t-1}$ denote the history before drawing (ξ_t, ζ_t) and write $\mathbb{E}_{t-1}[\cdot] := \mathbb{E}[\cdot \mid \mathcal{G}_{t-1}]$. Then

$$\mathbb{E}_{t-1}[w_t] = a(\gamma - 1) D_t.$$

The ξ_t-term is conditionally independent of the complete ζ_t-block, so Young's inequality $\|u+v\|_2^2 \le \lambda\|u\|_2^2 + \lambda(\lambda-1)^{-1}\|v\|_2^2$ gives

$$\mathbb{E}_{t-1}\|w_t\|_2^2 \le \lambda a^2 \delta_\gamma^2 \|x_t - x_{t-1}\|_2^2 + \beta^2 V_{\lambda,\tau}\sigma^2. \tag{59}$$

Expanding (58) under $\mathbb{E}_{t-1}$ and applying Young's inequality to the conditional-mean cross term gives

$$2a^2(1-\gamma)|\langle e_{t-1}, D_t\rangle| \le \beta a \|e_{t-1}\|_2^2 + \frac{a^3(1-\gamma)^2}{\beta}\|D_t\|_2^2.$$

Using $\|D_t\|_2 \le L\|x_t - x_{t-1}\|_2$, we therefore obtain

$$\mathbb{E}\|e_t\|_2^2 \le a\mathbb{E}\|e_{t-1}\|_2^2 + \beta^2 V_{\lambda,\tau}\sigma^2 + \left[\frac{a^3(1-\gamma)^2 L^2}{\beta} + \lambda a^2 \delta_\gamma^2\right] \mathbb{E}\|x_t - x_{t-1}\|_2^2. \tag{60}$$

Euclidean smoothness and (51) imply

$$f(x_{t+1}) - f_{\inf} \le f(x_t) - f_{\inf} - \frac{\eta}{2}\|F_t\|_2^2 + \frac{\eta}{2}\|e_t\|_2^2 + \left(\frac{L}{2} - \frac{1}{2\eta}\right)\|x_{t+1} - x_t\|_2^2. \tag{61}$$

For

$$H_t := f(x_t) - f_{\inf} + \frac{\eta}{2\beta}\|e_t\|_2^2,$$

combining (61) with (60) at time $t+1$ yields

$$\mathbb{E}[H_{t+1}] \le \mathbb{E}[H_t] - \frac{\eta}{2}\mathbb{E}\|F_t\|_2^2 + \frac{\eta\beta}{2}V_{\lambda,\tau}\sigma^2 + \frac{1}{2}\left(L - \frac{1}{\eta} + \eta A_\lambda\right)\mathbb{E}\|x_{t+1} - x_t\|_2^2. \tag{62}$$

Condition (56) gives $L\eta + \sqrt{A_\lambda}\eta \le 1$ and $A_\lambda\eta^2 \le \sqrt{A_\lambda}\eta$; hence $A_\lambda\eta^2 + L\eta \le 1$, so the last term is nonpositive. Telescoping (62), using $H_T \ge 0$, and averaging over $t = 0, \dots, T-1$ proves (57). □

For fixed $\lambda > 1$, the variance multiplier in (54) is minimized at

$$\tau_\lambda^* = \frac{\lambda}{2\lambda - 1}, \qquad V_{\lambda,\tau_\lambda^*} = \frac{\lambda}{2\lambda - 1}. \tag{63}$$

The endpoint $\lambda \downarrow 1$ is interpreted directly: taking $\tau = 1$ makes the fresh gradient independent of the correction and gives correction multiplier 1 and variance multiplier 1.

Corollary 2 (Matched proof-bound comparison)**.** *The coupled choices $1 < \Lambda < 2$ are Pareto-dominated by $\Lambda = 2$: both $\Lambda/\sqrt{\Lambda - 1}$ and $\Lambda/(\Lambda - 1)$ are no smaller there. It therefore suffices to fix $\Lambda \ge 2$. Analyze coupled MARS using the Young multiplier*

$$S_{\mathrm{M}} := \Lambda, \qquad V_{\mathrm{M}} := \frac{\Lambda}{\Lambda - 1}.$$

Analyze decoupled MARS using

$$\lambda := \frac{\Lambda}{2}, \qquad \tau := \frac{\Lambda}{2(\Lambda - 1)}, \tag{64}$$

with the limiting interpretation $(\lambda, \tau) = (1, 1)$ when $\Lambda = 2$. Then

$$S_{\mathrm{D}} = \frac{1}{2}S_{\mathrm{M}}, \qquad V_{\mathrm{D}} = \frac{1}{2}V_{\mathrm{M}}.$$

Consequently, after accounting for three stochastic-gradient evaluations per iteration for the decoupled method and two for coupled MARS, the coefficients produced by this proof for

$$\frac{\delta_\gamma \Delta\sigma}{\varepsilon^3} \quad \text{and} \quad \frac{(1-\gamma)L\Delta\sigma^2}{\varepsilon^4}$$

are both multiplied by $3/4$ relative to the matched coupled-MARS coefficients. The shared $L\Delta/\varepsilon^2$ coefficient is instead multiplied by $3/2$, so this is not a uniform total-complexity dominance.

Proof. Theorem 3 gives, up to universal numerical constants,

$$\beta \asymp \frac{\varepsilon^2}{V\sigma^2}, \qquad \eta^{-1} \lesssim L + \frac{(1-\gamma)L}{\beta} + \frac{\sqrt{S}\,\delta_\gamma}{\sqrt{\beta}}.$$

Thus, for a method using q stochastic-gradient evaluations per iteration, the two high-order oracle-complexity coefficients are proportional to

$$q\sqrt{SV} \quad \text{and} \quad qV,$$

respectively. For coupled MARS, $q_{\mathrm{M}} = 2$. By (63) and (64), the decoupled choice has $S_{\mathrm{D}} = S_{\mathrm{M}}/2$ and $V_{\mathrm{D}} = V_{\mathrm{M}}/2$, while $q_{\mathrm{D}} = 3$. Therefore

$$\frac{q_{\mathrm{D}}\sqrt{S_{\mathrm{D}}V_{\mathrm{D}}}}{q_{\mathrm{M}}\sqrt{S_{\mathrm{M}}V_{\mathrm{M}}}} = \frac{3}{4}, \qquad \frac{q_{\mathrm{D}}V_{\mathrm{D}}}{q_{\mathrm{M}}V_{\mathrm{M}}} = \frac{3}{4}.$$

The $L\Delta/\varepsilon^2$ term scales only with q, giving $3/2$. □

C CNN STUDY: PROTOCOL, TUNING GRIDS AND FULL TABLES

This appendix details the CNN protocol and gradient-budget accounting, reports the full tuning grids and results for three comparison regimes, and studies the sample-splitting weight τ at matched effective hyperparameters.

C.1 NETWORK AND GEOMETRY

All optimizers train the same convolutional network: the CifarNet architecture of the airbench CIFAR-10 benchmark (Jordan, 2024), in the version used for the CNN experiments of Scion (Pethick et al., 2025) and Gluon (Riabinin et al., 2025). A 2×2 whitening convolution with 24 channels, whose filters are set from the patch covariance of $5{,}000$ training images and then frozen (only its bias is trained), is followed by three convolutional groups of widths 64, 256 and 256, a final max-pooling layer, and a bias-free linear classifier whose output is divided by its input width. Each group applies a 3×3 convolution, 2×2 max-pooling, BatchNorm and a GELU activation, then a second 3×3 convolution, BatchNorm and GELU. The convolutions have no bias, and every BatchNorm keeps its scale frozen at one and trains only its bias.

VENUS and MARS use the same layer-wise Scion/Gluon LMO geometry (Pethick et al., 2025; Riabinin et al., 2025). Writing G for the momentum of a parameter tensor, the LMO direction of (2) is: for a convolution with C_{in} input channels, C_{out} output channels and kernel size k, the negative orthogonal polar factor of G reshaped to a $C_{\text{out}} \times C_{\text{in}}k^2$ matrix, scaled by $\sqrt{C_{\text{out}}/C_{\text{in}}}/k^2$; for a bias vector, $-G$ divided by its root mean square; and for the classifier head, $-\operatorname{sign}(G)$ divided by the head's input width. The step multiplies this direction by the stepsize and by a fixed radius of 1 for the convolutions and biases and 100 for the head, the values of unconstrained Scion for this network (Pethick et al., 2025, Table 10). The two methods share this update geometry, using the momentum estimators of Algorithms 1 and 2, respectively. AdamW is the standard Euclidean update on the same network.

C.2 IMPLEMENTATION AND COMPUTE

All CIFAR-10 runs use PyTorch (Paszke et al., 2019) (version 2.11 with CUDA 12.8) and one NVIDIA H200 GPU per run, scheduled on a Slurm cluster. Parameters, activations and gradients are stored in 32-bit floating point (float32); convolutions and matrix products use the TF32 tensor-core format, which keeps the float32 exponent range with a shorter mantissa.

The spectral LMO needs the polar factor $UV^\top$ of a matrix $G = U\Sigma V^\top$. Instead of computing an SVD, we approximate it by five steps of the Newton–Schulz iteration $X \leftarrow aX + b\,(XX^\top)X + c\,(XX^\top)^2X$, started from $X = G/(\|G\|_F + 10^{-7})$ and run in bfloat16. The classical Newton–Schulz iteration for the polar factor (Björck & Bowie, 1971; Higham, 2008) is the cubic case $(a, b, c) = (\frac{3}{2}, -\frac{1}{2}, 0)$; we use the quintic coefficients $(a, b, c) = (3.4445, -4.7750, 2.0315)$ of the Muon optimizer (Jordan et al., 2024b), as in Scion (Pethick et al., 2025) following Bernstein & Newhouse (2024). These coefficients give rapid approximate orthogonalization without requiring convergence to unit singular values. Both methods use the same Newton–Schulz implementation and iteration count.

Training images are augmented with random horizontal flips and random translations of up to two pixels. Recorded training times vary with the method and batch size, with a median of approximately 79 seconds per run.

C.3 TRAINING AND EVALUATION PROTOCOL

From the $50{,}000$-image CIFAR-10 training set we reserve a fixed $5{,}000$-image validation split with a fixed split seed (1234), held constant across all training seeds and independent of them; each run trains on the remaining $45{,}000$ images and the official $10{,}000$-image test set is never used for selection. We use a *constant* stepsize (no warmup, no decay) and label-smoothed cross-entropy ($\varepsilon = 0.2$). For the independently tuned comparisons, the summary tables report each run's maximum validation accuracy, the test accuracy at its corresponding checkpoint, and the minimum validation loss. The two accuracy metrics use six-view test-time augmentation (TTA), combining horizontal reflection with translated views; validation loss is single-view label-smoothed cross-entropy and can

Regime	Matched on	VENUS	MARS	AdamW
I (equal gradients/step)	grads/step, steps	$b1000$, p=32	$b1500$, p=48	$b3000$, p=96
II (equal batch, large)	batch 1000	$b1000$, p=32	$b1000$, p=48	$b1000$, p=96
III (equal batch, small)	batch 128	$b128$, p=32	$b128$, p=48	$b128$, p=96

Table 3: Gradient-budget matching for the three regimes, with a nominal budget of 96 gradient-work epochs (4.32M per-example gradients). "grads/step" is batch size times the number of minibatch-gradient evaluations per step; p denotes data passes per stream. VENUS uses two independent streams. Actual budgets account for dropped incomplete minibatches (Section C.4).

attain its minimum at a different checkpoint. Table entries are means ± standard errors of the mean (SEM) of these per-run quantities. Test metrics are recorded for reporting and do not enter checkpoint or configuration selection.

Validation is evaluated at fixed step intervals and at epoch boundaries. The logged curve runs have 160, 144 and 192 checkpoints for VENUS, MARS and AdamW, respectively, in Regime I; 160, 240 and 192 in Regime II; and 172, 258 and 192 in Regime III. Selecting maxima or minima over checkpoints introduces selection bias; matching checkpoint counts alone would not ensure equal bias because variability and dependence across checkpoints also matter.

C.4 Gradient-budget accounting

For batch size b and K optimizer steps, the gradient budget is $W = gKb$, where $g = 3$ for VENUS, $g = 2$ for MARS and $g = 1$ for AdamW. In VENUS, each of the independent ξ and ζ streams has batch size b, with the ζ gradient evaluated at both iterates. We target $W = 4.32$M per-example gradients: 32 data passes per stream for VENUS, 48 for MARS and 96 for AdamW. Incomplete minibatches are dropped, so the actual budget is 4,313,088 at batch 128 and 4,276,224 at batch 512 in the τ study; it is exactly 4.32M at batches 1000 and 1500 and for AdamW at batch 3000. Budgets remain matched across methods within each regime. The horizontal axis of the learning curves is W/N, with $N = 45{,}000$; one plotted *epoch* therefore denotes N per-example gradient evaluations. This accounting matches gradient work, but not sample exposure, evaluation cost or wall-clock time. Table 3 lists the three regimes; Regime I also matches optimizer-step counts.

C.5 Tuning pipeline, search grids and best configurations

Each optimizer is tuned independently for each batch size at the corresponding gradient budget. For VENUS and MARS, coarse grids are extended to bracket the selected learning rates. For AdamW, following Semenov et al. (2025); Orvieto & Gower (2025), we fix $\varepsilon = 10^{-8}$ and global gradient-norm clipping at 1.0, and sweep the learning rate, momentum pair and weight decay. Table 4 gives the complete grids and selected configurations. Across methods, batches and seeds, the archived study contains over twenty thousand runs.

Initial grid searches use seed 0. Final configurations for VENUS at batch 1000 and MARS at batch 1500 are selected by the mean of the per-seed best validation accuracies over 50 seeds, among 19 and 10 candidate configurations, respectively. At batch 128, two VENUS candidates are compared over 20 seeds using the same criterion. These multi-seed evaluations therefore contribute to configuration selection as well as to the reported results. The batch-1000 VENUS configuration is reused in Regimes I and II. The result tables use 50 seeds for VENUS at batch 1000 and MARS at batch 1500, and 20 seeds for the other configurations.

Training budgets are matched per run, but hyperparameter-search budgets differ across methods (Table 4). The independently tuned results should therefore be interpreted together with the comparison at matched effective hyperparameters in Section C.7.

Learning-rate response. The selected learning rates are interior to their grids. In Regime I they are 2^{-3} for VENUS and 2^{-4} for MARS, with worse adjacent grid values on both sides. The initial AdamW optimum at batch 3000 was on the upper boundary, so we extended its grid; the selected $\eta = 0.1$ then outperformed 0.3 and 1.0. These checks bracket the selected rates without establishing

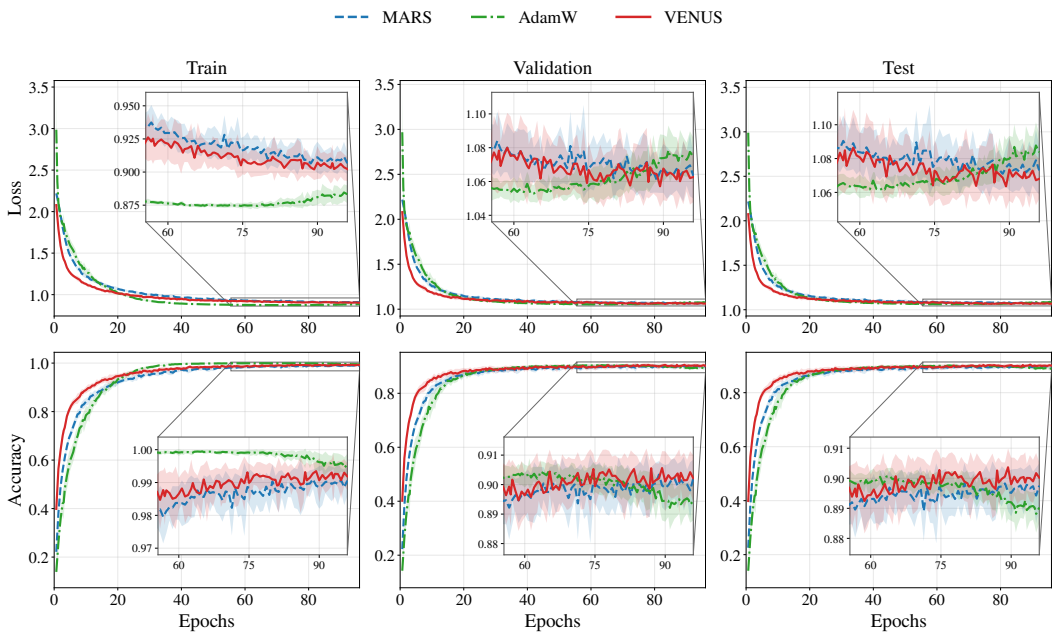


Figure 6: Regime I (equal gradients per step): VENUS at batch 1000, MARS at 1500, AdamW at 3000, all at 96 epochs (4.32M per-example gradients). Six unaugmented single-view learning curves (columns: train / validation / test; rows: loss / accuracy) against epochs, mean over 20 seeds with $\pm$ one standard-deviation bands; each inset shows late training. The horizontal axis counts gradient work, not data passes, so the three methods end at the same position (Section C.4).

equal search coverage. AdamW selects zero weight decay in every regime and $\eta = 0.003$ at batches 128 and 1000, compared with 0.1 at batch 3000.

Parameter convention. We report selected MARS configurations as (β, γ, η) and selected VENUS configurations as $(\beta, \gamma, \tau, \eta)$, consistently with Section 3.1. The CNN VENUS search was performed in the primitive coordinates (μ, p, θ, η). Its evaluated configurations map to effective coordinates through

$$\beta = (1-\theta)\mu + \theta p, \qquad \gamma = \frac{\theta(1-p)}{1-\beta}, \qquad \tau = \frac{(1-\theta)\mu}{\beta},$$

with η unchanged. The image of the original Cartesian grid is a non-Cartesian set in effective coordinates; it must not be interpreted as independent sweeps over β, γ, and τ. Table 4 retains the original grid coordinates for reproducibility and reports the selected VENUS configurations in both primitive and effective coordinates. This conversion changes neither the evaluated configurations nor their results.

C.6 Learning curves and per-regime behavior

Each figure shows single-view training, validation and test loss and accuracy, averaged over 20 seeds with one-standard-deviation bands and insets of late training. Training metrics use a fixed 5,000-image training subset; validation and test metrics use their full splits. The summary tables instead use checkpoint-selected metrics and TTA accuracies, and some entries average a larger collection of runs. These differences in evaluation and aggregation can change the ordering when performance gaps are small.

Regime I (equal gradients per step). In Figure 6, VENUS improves accuracy more rapidly early in training: by the eleventh gradient-work epoch, its mean validation and test accuracies exceed those of MARS by about three percentage points and AdamW by about eight. It also has the best mean values on all three summary metrics in Table 1, although the mean learning curves cross later in training.

Method	Batch	Original swept values	#cfg	Best configuration
VENUS	1000	$\mu \in \{0.001, 0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3\}$ (13) $p \in \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$ (12) $\theta \in \{0.001, 0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3\}$ (13) $\eta \in \{2^0, 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}\}$ (7)	14,196	$(\mu, p, \theta, \eta)=$ $(0.1, 0.95, 0.1, 0.125)$ $(\beta, \gamma, \tau, \eta)=$ $(0.185, 0.006135, 0.486486, 0.125)$
	128	$\mu \in \{0.05, 0.075, 0.1, 0.15, 0.2\}$ (5) $p \in \{0.6, 0.7, 0.8, 0.9, 0.95\}$ (5) $\theta \in \{0.001, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35\}$ (8) $\eta \in \{2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}\}$ (6)	1,200	$(\mu, p, \theta, \eta)=$ $(0.1, 0.8, 0.25, 0.0625)$ $(\beta, \gamma, \tau, \eta)=$ $(0.275, 0.068966, 0.272727, 0.0625)$
MARS	1500	$\beta \in \{0.001, 0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55\}$ (18) $\gamma \in \{0.001, 0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3, 0.35, 0.4\}$ (15) $\eta \in \{2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}\}$ (5)	1,350	$(\beta, \gamma, \eta)=$ $(0.075, 0.001, 0.0625)$
	1000	$\beta \in \{0.05, 0.075, 0.1, 0.15, 0.2\}$ (5) $\gamma \in \{0.001, 0.05, 0.1\}$ (3) $\eta \in \{2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}\}$ (4)	60	$(\beta, \gamma, \eta)=$ $(0.1, 0.05, 0.0625)$
	128	$\beta \in \{0.001, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2\}$ (7) $\gamma \in \{0.001, 0.05, 0.1\}$ (3) $\eta \in \{2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}\}$ (6)	126	$(\beta, \gamma, \eta)=$ $(0.05, 0.001, 0.03125)$
AdamW	3000	$\eta \in \{10^{-4}, 3{\cdot}10^{-4}, 10^{-3}, 3{\cdot}10^{-3}, 10^{-2}, 3{\cdot}10^{-2}, 10^{-1}, 0.3, 1\}$ (9) $(\beta_1, \beta_2) \in \{(0.9, 0.9), (0.95, 0.95), (0.975, 0.975), (0.9, 0.999)\}$ (4) $\lambda \in \{0, 0.1\}$ (2)	72	$(\eta, \beta_1, \beta_2, \lambda)=$ $(0.1, 0.9, 0.999, 0)$
	1000	$\eta \in \{10^{-4}, 3{\cdot}10^{-4}, 10^{-3}, 3{\cdot}10^{-3}, 10^{-2}, 3{\cdot}10^{-2}, 10^{-1}\}$ (7) $(\beta_1, \beta_2) \in \{(0.9, 0.9), (0.95, 0.95), (0.975, 0.975), (0.9, 0.999)\}$ (4) $\lambda \in \{0, 0.1\}$ (2)	56	$(\eta, \beta_1, \beta_2, \lambda)=$ $(0.003, 0.9, 0.999, 0)$
	128	$\eta \in \{10^{-4}, 3{\cdot}10^{-4}, 10^{-3}, 3{\cdot}10^{-3}, 10^{-2}, 3{\cdot}10^{-2}, 10^{-1}\}$ (7) $(\beta_1, \beta_2) \in \{(0.9, 0.9), (0.95, 0.95), (0.975, 0.975), (0.9, 0.999)\}$ (4) $\lambda \in \{0, 0.1\}$ (2)	56	$(\eta, \beta_1, \beta_2, \lambda)=$ $(0.003, 0.9, 0.9, 0)$

Table 4: Original search grids and selected configurations for each method and batch. For VENUS, best configurations are shown in both primitive (μ, p, θ, η) and effective $(\beta, \gamma, \tau, \eta)$ coordinates; effective γ and τ are rounded to six decimal places. MARS uses (β, γ, η) and AdamW uses $(\eta, \beta_1, \beta_2, \lambda)$. The original VENUS grid uses primitive branch weights. The nominal budget is 96 gradient-work epochs (Section C.4), with 32 data passes per stream for VENUS, 48 for MARS and 96 for AdamW; the number of values on each axis is given in parentheses. Every grid here is a full Cartesian product: all combinations of the listed values were run, so #cfg is the product of the numbers in parentheses. For AdamW the pair (β_1, β_2) is a single axis with four values.

Regime II (equal batch, large). At batch 1000, AdamW generally achieves better single-view metrics in late training (Figure 7). The VENUS–MARS ordering varies across metrics and checkpoints; in particular, MARS has lower mean single-view validation loss over substantial portions of training.

Metric	VENUS (decoupled)	MARS (coupled)	AdamW ($\lambda = 0$)
Validation accuracy ↑	$\mathbf{0.9310 \pm 0.0003}$	0.9292 ± 0.0005	0.9298 ± 0.0003
Min. validation loss ↓	1.0376 ± 0.0006	1.0394 ± 0.0008	$\mathbf{1.0325 \pm 0.0010}$
Test acc. at best val ↑	$\mathbf{0.9241 \pm 0.0005}$	0.9212 ± 0.0006	0.9209 ± 0.0004

Table 5: Large batch, **equal batch per step** (batch 1000): VENUS, MARS and AdamW all at batch 1000 and 4.32M per-example gradients. Mean ± SEM (VENUS: 50 seeds; MARS/AdamW: 20 seeds); best mean in bold.

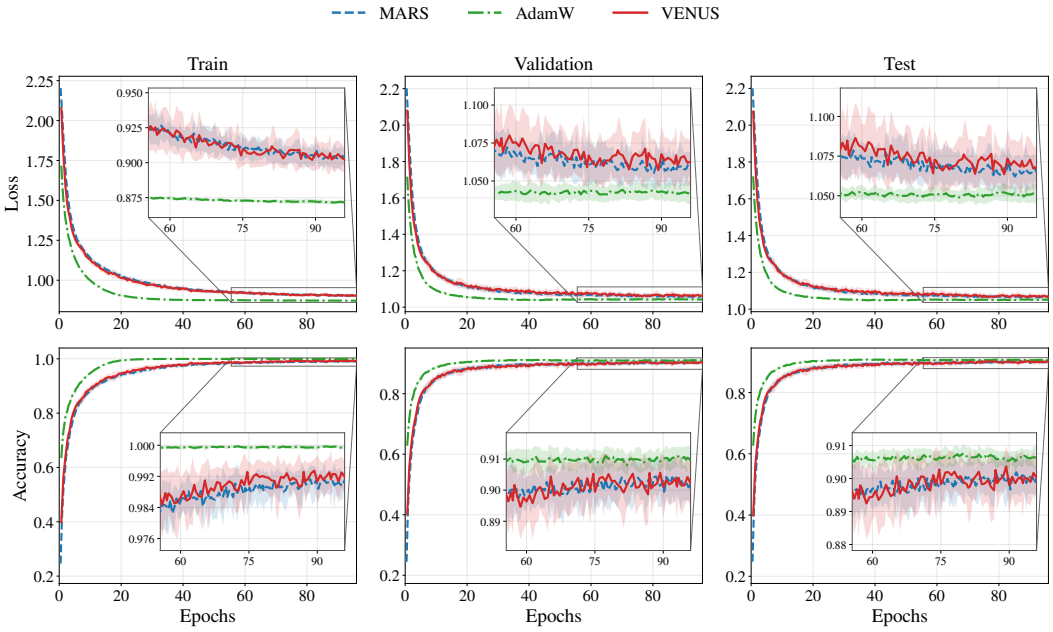


Figure 7: Regime II (equal batch 1000): VENUS, MARS and AdamW all at batch 1000 and 96 gradient-work epochs. Layout as in Figure 6. AdamW generally achieves lower loss and higher accuracy in late training. At the matched gradient budget, it takes three times as many optimizer steps as VENUS; this comparison does not isolate the effect of step count.

The summary table instead gives VENUS the highest mean validation and test accuracies and AdamW the lowest reported validation loss (Table 5).

Regime III (equal batch, small). At batch 128, AdamW achieves the lowest mean losses in late training and the best mean values in Table 6 (Figure 8). VENUS has higher mean validation accuracy and lower reported validation loss than MARS. Their mean test-accuracy difference is only 0.00023, small relative to the SEM of approximately 0.00053 for each method; these results do not establish a clear test-accuracy improvement.

Summary of CNN results. The selected VENUS configurations have favourable mean summary metrics relative to MARS, with the largest differences in Regime I. The learning curves do not show uniform dominance, and the test-accuracy gap at batch 128 is small relative to its across-seed uncertainty. At equal batch sizes, AdamW generally achieves better single-view metrics in late training. These comparisons vary optimizer-step count and sample exposure alongside the estimator, so they do not identify the mechanism responsible for the observed ordering.

Metric	VENUS (decoupled)	MARS (coupled)	AdamW ($\lambda = 0$)
Validation accuracy ↑	0.9356 ± 0.0003	0.9339 ± 0.0004	$\mathbf{0.9368 \pm 0.0004}$
Min. validation loss ↓	1.0288 ± 0.0006	1.0347 ± 0.0009	$\mathbf{1.0187 \pm 0.0009}$
Test acc. at best val ↑	0.9278 ± 0.0005	0.9276 ± 0.0005	$\mathbf{0.9283 \pm 0.0004}$

Table 6: Small batch, **equal batch per step** (batch 128): VENUS, MARS and AdamW all at batch 128 and a nominal budget of 4.32M per-example gradients (actual: 4,313,088). Mean ± SEM (20 seeds each); best mean in bold.

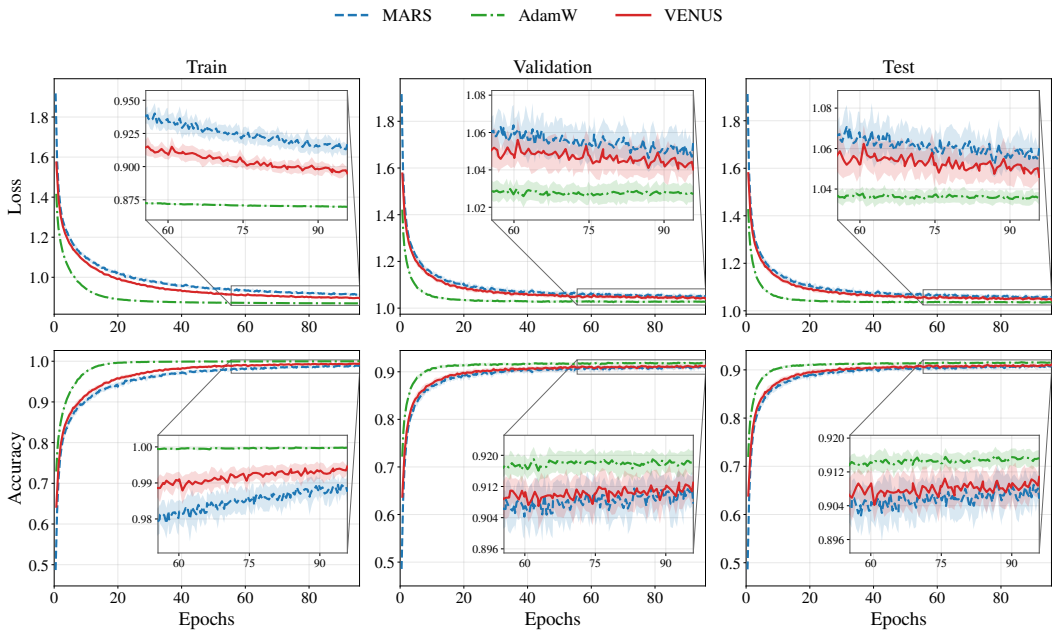


Figure 8: Regime III (equal batch 128): VENUS, MARS and AdamW at a nominal budget of 96 gradient-work epochs, with 4,313,088 per-example gradients actually evaluated. Layout as in Figure 6. AdamW generally achieves the best late-training single-view metrics. VENUS has lower mean losses than MARS, while their accuracy curves occasionally cross.

C.7 Sensitivity to τ at matched settings

The selected VENUS configurations correspond to $\tau \approx 0.49$ at batch 1000 and $\tau \approx 0.27$ at batch 128, but the primitive-coordinate search does not isolate the effect of τ. We therefore compare MARS and VENUS at identical effective (β, γ, η), varying only the sample-splitting weight within VENUS.

Design. At batches 128, 512, 1000 and 1500, we test $\beta \in \{0.025, 0.05, 0.1, 0.2, 0.3\}$ and $\gamma \in \{0.001, 0.05, 0.1\}$. The stepsizes are $\{2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}\}$ at batch 128, $\{2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}\}$ at batch 512, and $\{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}\}$ at batches 1000 and 1500. Each configuration is evaluated for MARS and for VENUS with $\tau \in \{0.1, 0.5, 0.75\}$, using the same three seeds (0, 1 and 2) and the gradient budgets of Section C.4. For each run, we average each single-view metric over the last $\max(1, \lfloor 0.1T \rfloor)$ logged checkpoints, where T is the number of checkpoints, and then average across seeds. This defines a tail average over records, not an exact final fraction of the gradient budget. VENUS settings are converted to primitive weights for execution; both parameterizations are logged.

Stability at large stepsizes. We classify a method–configuration pair as *unstable* when its tail validation loss, averaged over the three seeds, exceeds 1.5. At batches 128, 512, 1000 and 1500, respectively, MARS exceeds this threshold in $15/60$, $17/75$, $11/60$ and $9/60$ configurations, compared with $7/60$, $10/75$, $5/60$ and $5/60$ for VENUS at $\tau = 0.5$. Most occur at the largest tested stepsize:

Batch	τ	n	Train loss	Validation loss	Test loss	Test accuracy
128	0.1	45	+0.0022 (19)	−0.0042 (32)	−0.0030 (29)	+0.0017 (31)
	0.5		−0.0131 (45)	−0.0093 (40)	−0.0072 (42)	+0.0027 (36)
	0.75		−0.0080 (44)	−0.0072 (34)	−0.0063 (40)	+0.0029 (39)
512	0.1	58	+0.0042 (26)	−0.0035 (34)	−0.0011 (30)	+0.0012 (35)
	0.5		−0.0104 (57)	−0.0079 (46)	−0.0061 (41)	+0.0023 (34)
	0.75		−0.0067 (41)	−0.0071 (39)	−0.0051 (36)	+0.0023 (39)
1000	0.1	49	−0.0025 (26)	−0.0109 (36)	−0.0082 (35)	+0.0048 (40)
	0.5		−0.0162 (46)	−0.0143 (44)	−0.0137 (41)	+0.0066 (39)
	0.75		−0.0114 (41)	−0.0130 (38)	−0.0125 (39)	+0.0064 (41)
1500	0.1	51	+0.0021 (22)	−0.0075 (35)	−0.0058 (35)	+0.0038 (37)
	0.5		−0.0130 (48)	−0.0113 (43)	−0.0090 (40)	+0.0044 (40)
	0.75		−0.0092 (37)	−0.0116 (39)	−0.0105 (38)	+0.0061 (38)

Table 7: Paired study of τ on CIFAR-10. Entries give the median VENUS-minus-MARS difference across configurations and, in parentheses, the number favouring VENUS. Loss differences favour VENUS when negative; accuracy differences when positive. Per-configuration metrics average the last 10% of logged checkpoints within each run, then the three shared seeds. All τ values use the same n configurations, retained when mean tail validation loss is at most 1.5 for MARS and all three VENUS variants. There are 60 configurations per batch (75 at batch 512) before filtering.

12, 12, 9 and 8 of the 15 configurations at that stepsize for MARS, versus 7, 9, 5 and 5 for VENUS. Thus VENUS at $\tau = 0.5$ has fewer high-loss configurations under the tested grid and threshold.

Paired comparison. Figures 10–13 and Table 7 use the common subset whose mean tail validation loss is at most 1.5 for MARS and all three VENUS variants. Each plotted point is one (β, γ, η) configuration, with MARS on the horizontal axis and VENUS on the vertical axis, after averaging over three shared seeds. Red points indicate lower loss (below the diagonal) or higher accuracy (above it) for VENUS.

Among the tested values, $\tau = 0.5$ gives the largest number of favourable comparisons in 13 of 16 batch–metric combinations and the most favourable median difference in 12 of 16. The median differences are least favourable at $\tau = 0.1$ throughout; its median training-loss difference favours MARS at three of the four batch sizes. At $\tau = 0.5$ and 0.75, median validation-loss differences range from -0.0143 to -0.0071 and median test-accuracy differences from $+0.0023$ to $+0.0066$. These medians favour VENUS at every tested batch size within the retained configurations, without implying improvement at every configuration.

Response to τ at a fixed configuration. At one fixed configuration per batch, we evaluate $\tau \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$ on five fresh seeds (10–14), using the same tail average (Figure 9). Across these values, the mean validation-loss range is 0.0041, 0.0038, 0.0089 and 0.0091 at batches 128, 512, 1000 and 1500, respectively. The lowest observed mean occurs at $\tau = 0.5$ for batches 128 and 1500, at 0.25 for batch 512, and at 0.75 for batch 1000; $\tau = 0.5$ remains within approximately 0.0038 of the lowest mean at each batch. The dashed MARS reference uses the three grid seeds, so its difference from the five-seed VENUS curve is not a paired comparison.

Interpretation. The matched-setting grid and fixed-configuration responses support $\tau = 0.5$ as a practical default. This choice also minimizes the explicit factor $\chi_\tau = \sqrt{\tau^2 + (1-\tau)^2}$ in Section 3.2.2, but these results do not establish a universal optimum or rule out improvements from tuning τ.

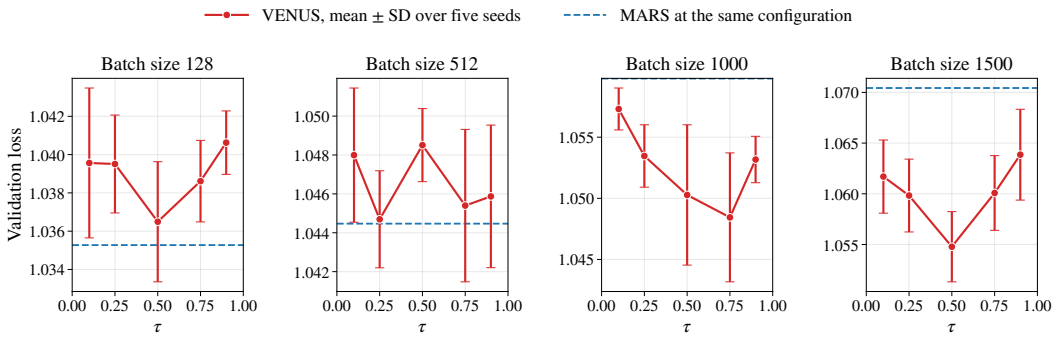


Figure 9: Response to τ at one fixed configuration per batch, on five fresh seeds. The configurations are $(\beta, \gamma, \eta) = (0.05, 0.05, 2^{-6})$ at batch 128, $(0.05, 0.05, 2^{-5})$ at batch 512, $(0.1, 0.05, 2^{-4})$ at batch 1000 and $(0.05, 0.001, 2^{-4})$ at batch 1500. Error bars are ± 1 standard deviation of the per-run tail validation losses over the five seeds. The dashed line is MARS at the same configuration, from the three grid seeds; it is not a paired comparison with the VENUS curve. Each panel has its own scale.

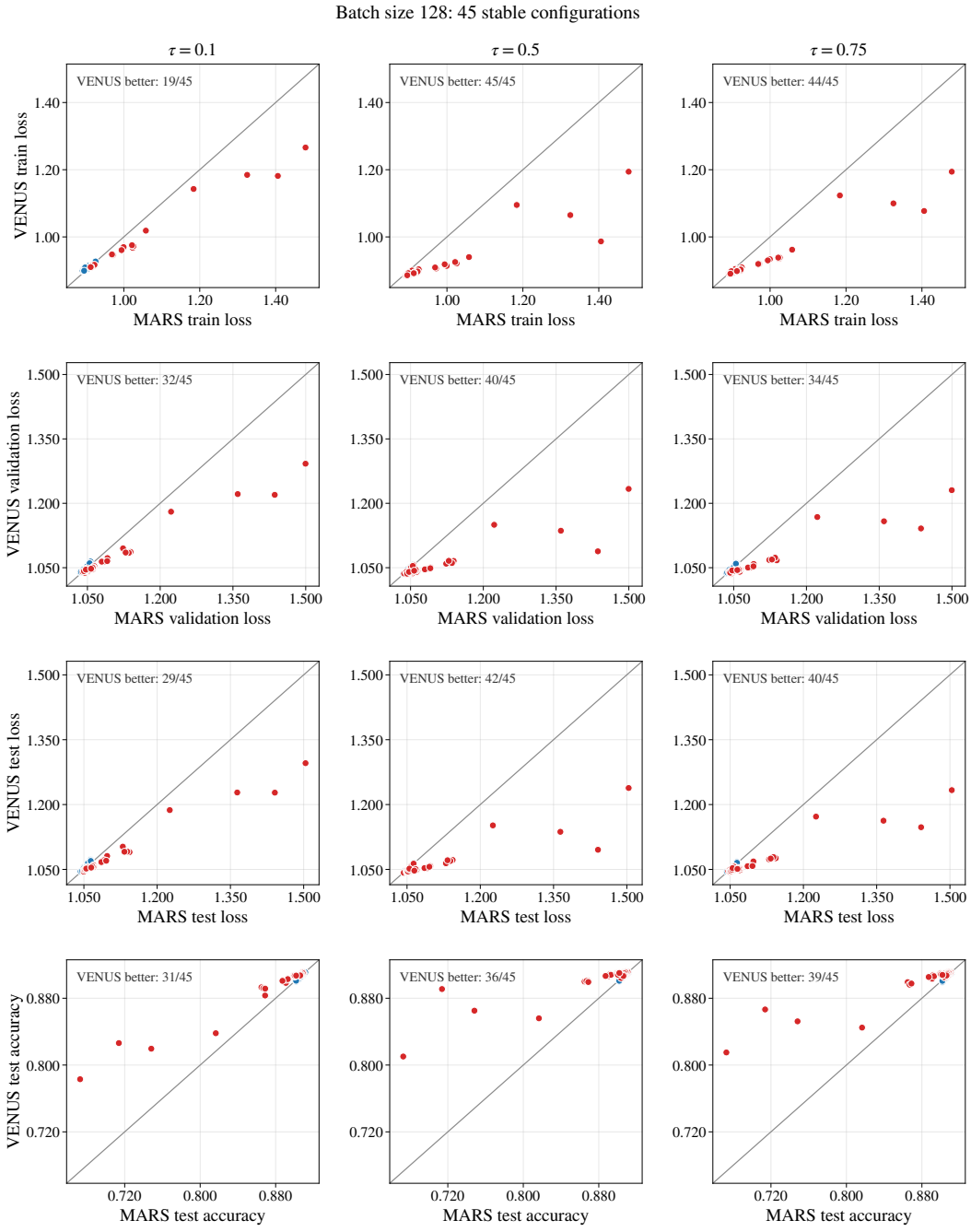


Figure 10: Batch 128: VENUS against MARS at identical (β, γ, η), retaining the common configurations whose mean tail validation loss is at most 1.5 for MARS and all three VENUS variants. Rows show training, validation and test loss and test accuracy; columns show the three τ values. Red points indicate lower loss or higher accuracy for VENUS. Each point averages the last 10% of logged checkpoints and three shared seeds, as in Table 7. All three panels of a row share their scale.

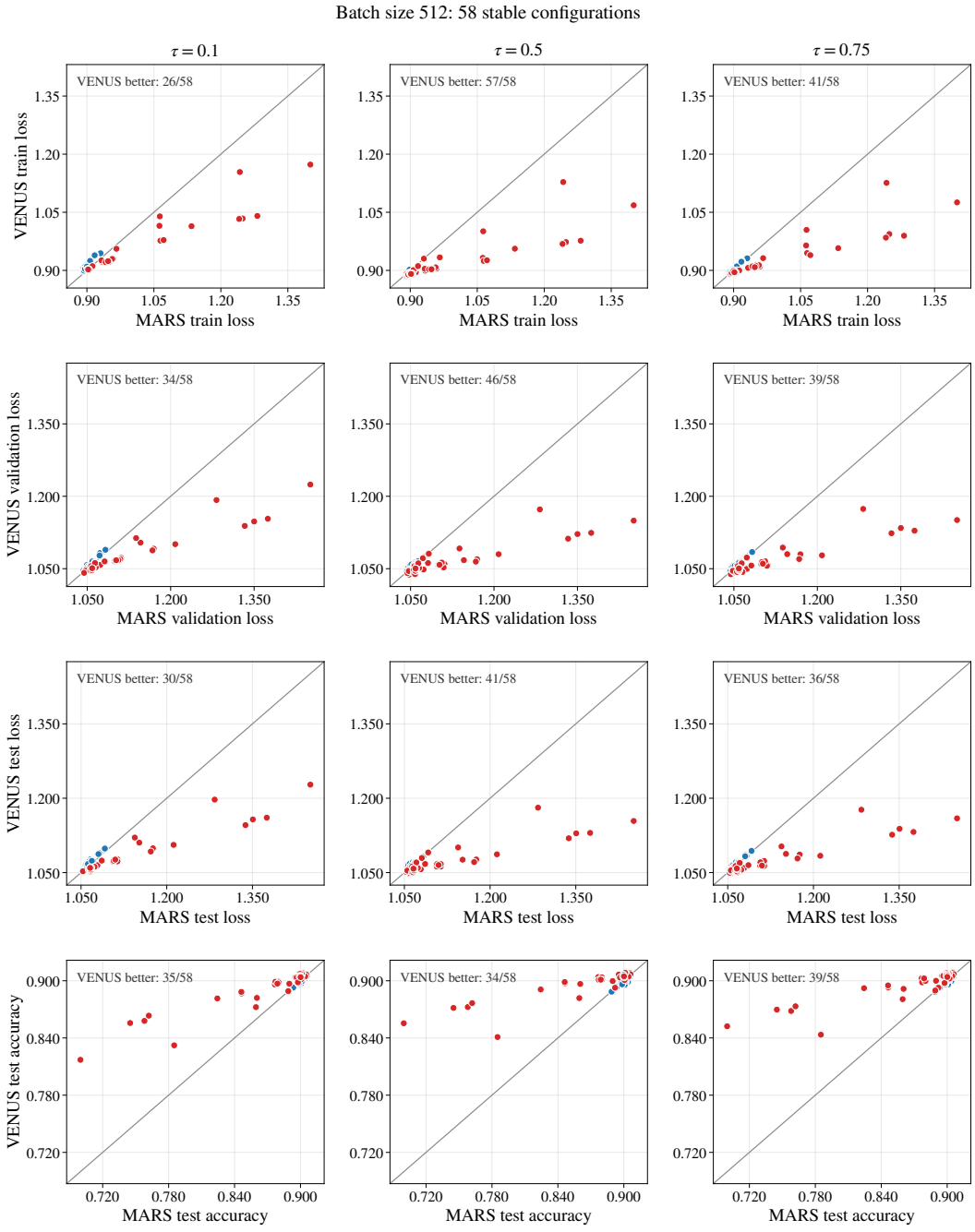


Figure 11: Batch 512, as in Figure 10.

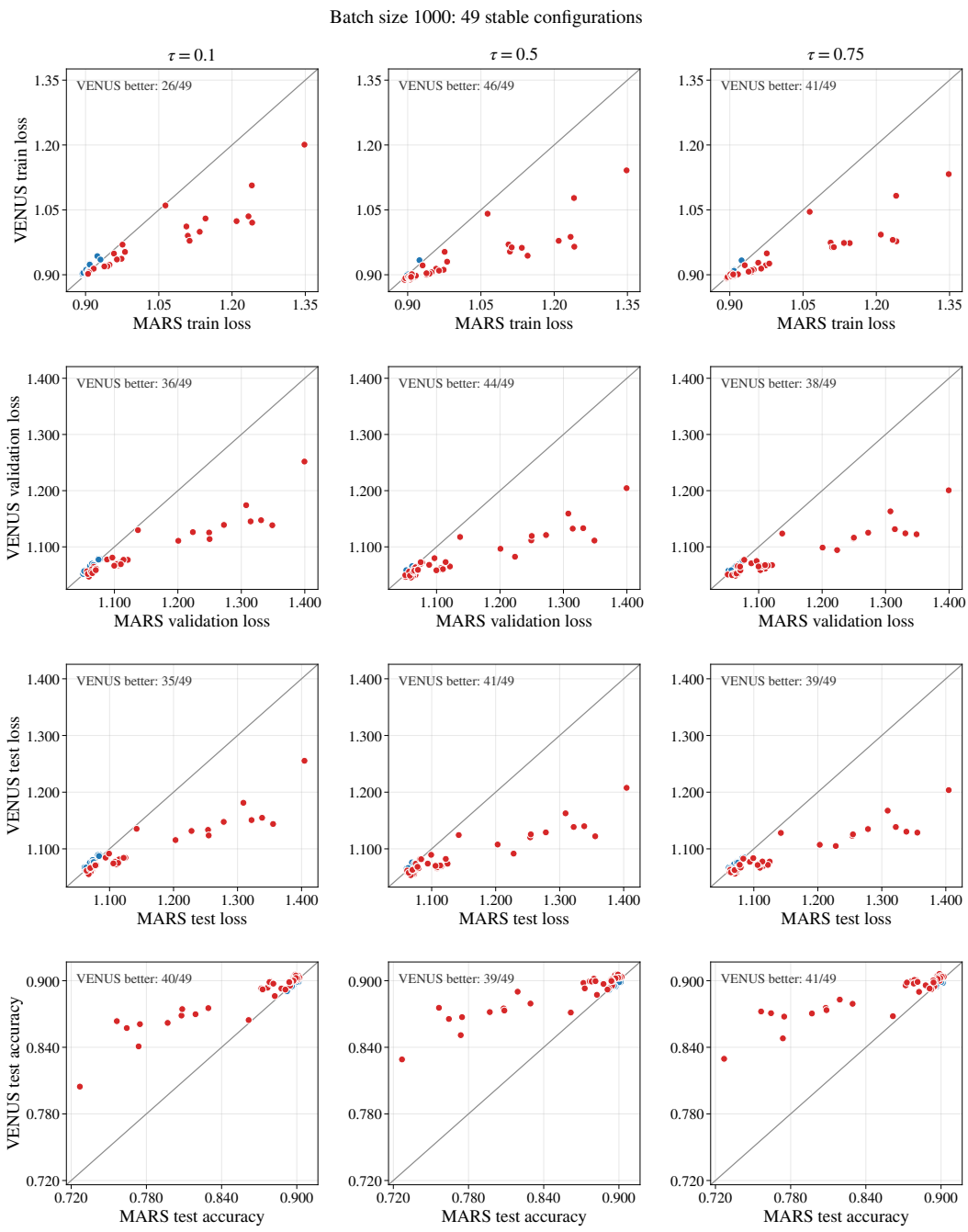


Figure 12: Batch 1000, as in Figure 10.

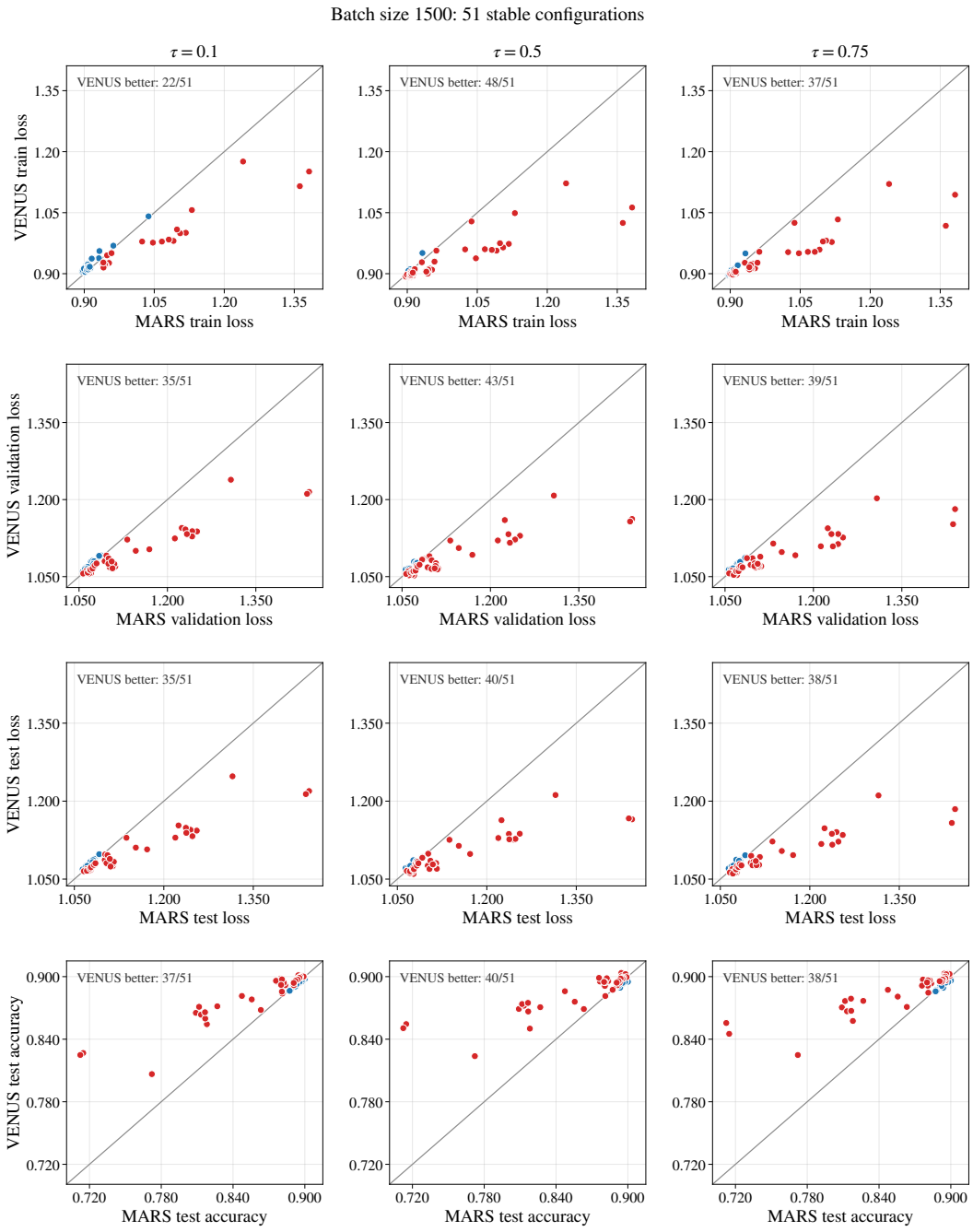


Figure 13: Batch 1500, as in Figure 10.

D LLM STUDY: PROTOCOL, TUNING AND FULL RESULTS

We compare exact MARS and VENUS at batch sizes 16, 32 and 64, and their cached variants at batch sizes 16 and 32. The primary outcome is final validation loss at a matched gradient-token budget.

Scope. Here, *exact* means that both endpoints of the transport difference are evaluated on the same minibatch, as analysed in Section 3.2.2; it does not imply a full-data gradient or an exact LMO. The matrix update uses Newton–Schulz orthogonalization. Cached variants reuse the previous iteration's gradient and are reported separately in Section D.5, outside the scope of that analysis. Small-batch training under limited resources motivates the study, but equal gradient-token work does not imply equal FLOPs, memory, data exposure or wall-clock time.

D.1 MODEL, DATA AND HARDWARE

Model. The network is the 124M-parameter GPT-style decoder-only Transformer of the NanoGPT experiments of Scion (Pethick et al., 2025, Table 7) and Gluon (Riabinin et al., 2025), which build on the modded-nanogpt codebase (Jordan et al., 2024a); our implementation is adapted from the code of Gruntkowska et al. (2025). It has 12 Transformer blocks with residual width 768. Each block adds two residual branches, causal self-attention and an MLP, and each branch starts with an RMS normalization (Zhang & Sennrich, 2019) without learnable parameters (pre-normalization).

- *Attention.* 6 heads of dimension 128. In every head, queries and keys are RMS-normalized (Henry et al., 2020) and rotated by rotary position embeddings (Su et al., 2024) with base 10,000; the attention is causal, with the usual $1/\sqrt{128}$ scaling. The query, key, value and output projections are bias-free 768×768 matrices.
- *MLP.* A bias-free map from width 768 to 3,072, the squared-ReLU activation $x \mapsto 2\max(x, 0)^2$ (So et al., 2021), and a bias-free map back to width 768.
- *Embedding and output.* Tokens enter through an embedding matrix of size $50{,}304 \times 768$ (the GPT-2 vocabulary, padded to 50,304 entries); there is no learned positional embedding. After the last block, a final RMS normalization and an output head tied to the embedding produce the logits.

The model has no biases, no learnable normalization parameters and no dropout. Of its 123.6M parameters, 84.9M are the 72 matrices of the blocks and 38.6M the tied embedding. The query, key, value and MLP input matrices are initialized uniformly with standard deviation $0.5/\sqrt{768}$, the output projections of both branches of every block at zero, and the tied embedding with the default PyTorch initialization of a linear layer. Parameters are stored in float32; the blocks run their forward and backward passes in bfloat16, and the final normalized activations are cast to float32 before the tied output head.

Data. We train on the 10B-token sample of FineWeb (Penedo et al., 2024), tokenized with the GPT-2 tokenizer, in sequences of 1,024 tokens. Batch size B is the number of sequences per gradient evaluation; both VENUS minibatches, ξ and ζ, contain B sequences. Loss is mean next-token cross-entropy, using natural logarithms. Validation uses the separate FineWeb validation shard (Section D.2).

Parameter groups. MARS or VENUS updates the 72 matrices of the Transformer blocks with the spectral-norm LMO of (2). The LMO approximates the orthogonal polar factor of the momentum with the same five-step quintic Newton–Schulz iteration as the CNN study (Section C.2; Jordan et al., 2024b), run in bfloat16, and scales it by $\sqrt{d_{\text{out}}/d_{\text{in}}}$ and by a fixed radius of 50, the value of Scion for this model (Pethick et al., 2025, Table 7); this group uses no weight decay. The tied embedding, the only parameter outside the blocks, is trained with AdamW as in Muon (Jordan et al., 2024b) rather than with the sign-type LMO of Scion: stepsize 3×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.95)$ and weight decay 0.1, identical in every run. Only the MARS/VENUS group is tuned. No gradient clipping is used.

Hardware and software. Each run uses one NVIDIA H200 GPU, without data parallelism, and accumulates gradients over micro-batches of 8 sequences. The code uses PyTorch (Paszke et al.,

2019) (version 2.11 with CUDA 12.8), and runs are scheduled on a Slurm cluster. Median runtimes range from 2h55m to 3h50m across method–batch combinations (Table 8). Logged training time totals approximately 1,700 GPU-hours across 542 completed runs, excluding validation overhead and terminated runs.

D.2 Training and validation protocol

Budget and schedule. Each run is allocated 2,699,034,624 gradient tokens, which determines its planned number of optimizer steps K (Section D.3); runs terminated for divergence do not exhaust this budget. At step $k = 0, \ldots, K-1$, both parameter groups multiply their peak stepsize, η for MARS/VENUS and 3×10^{-4} for AdamW, by the same factor

$$s_k = \begin{cases} \dfrac{k+1}{W}, & 0 \le k < W, \\ 0.1 + 0.9 \cdot \dfrac{1}{2}\left(1 + \cos\dfrac{\pi\,(k-W)}{K-W-1}\right), & W \le k \le K-1, \end{cases}$$

where W is 1% of K, rounded to the nearest integer. The factor thus rises linearly to 1 over the first 1% of the steps and then decreases along a half cosine to 0.1, which it reaches at the last step.

Validation. Every run uses the same fixed validation set: 200 batches of 8 sequences (1.64M tokens) from the FineWeb validation shard, in a fixed order with seed 1234. Validation is evaluated at 12 checkpoints spaced every 224.9M gradient tokens, plus an initial check after 500 steps. A run is classified as diverged and terminated if any validation loss is non-finite or exceeds 10.

Selection and confirmation. Hyperparameters are selected by final validation loss on seed 52, then confirmed on six fresh seeds 53–58. For a given seed and batch size, methods share the model initialization and the ζ minibatch stream; VENUS uses a separately seeded ξ stream. We define the paired difference $d_s = \mathcal{L}_{\mathrm{VENUS}}(s) - \mathcal{L}_{\mathrm{MARS}}(s)$, so negative values favour VENUS, and report its mean $\bar{d}$, sample standard deviation s_d and 95% Student-t interval $\bar{d} \pm t_{0.975,5} s_d/\sqrt{6}$. Paired t-tests and sign tests are two-sided and unadjusted for multiple comparisons. Other pairwise differences follow the ordering stated in the corresponding table.

For the selected exact configurations at batch sizes 16, 32 and 64, the Pearson correlations between methods across seeds are 0.33, 0.84 and 0.78, respectively. Across these six configurations, mean confirmation loss minus tuning-seed loss ranges from -0.0008 to $+0.0025$; these observed differences are not bounds on selection bias.

D.3 Compute accounting

One *gradient token* is one token participating in one stochastic-gradient evaluation. Exact MARS evaluates two gradients per step on the same minibatch ζ; exact VENUS adds a third on the independent minibatch ξ. Their cached variants evaluate one and two gradients per step, respectively. At batch size B, each evaluation processes $1{,}024B$ tokens, so the budget fixes the update counts in Table 8.

Gradient-token work is a proxy for computational cost. Median runtimes for exact MARS, exact VENUS and approximate VENUS differ by less than 7% within each batch size where they were run. Approximate MARS takes approximately 16% longer than exact MARS at batch size 16 and 9% longer at batch size 32; its additional optimizer and LMO operations are not included in gradient-token accounting.

The table also reports sampled training tokens, counting each minibatch once regardless of recomputation. These counts differ across methods and do not remove overlap between independently sampled streams; they therefore do not measure globally unique tokens.

D.4 Exact estimators: MARS and VENUS

This section compares the two exact estimators at batch sizes 16, 32 and 64. Each method is tuned independently at every batch size, and the selected configurations are confirmed on six paired seeds.

Method	Grads/step	Optimizer steps			Sampled training tokens	Wall-clock time per run		
		B=16	B=32	B=64		B=16	B=32	B=64
MARS	2	82,368	41,184	20,592	1.35B	3h19m	3h04m	2h57m
VENUS	3	54,912	27,456	13,728	1.80B	3h09m	3h00m	2h55m
approximate MARS	1	164,736	82,368	–	2.70B	3h50m	3h20m	–
approximate VENUS	2	82,368	41,184	–	2.70B	3h21m	3h06m	–

Table 8: Compute accounting at 2.70B gradient tokens per run. The number of sampled training tokens does not depend on the batch size. Wall-clock time is the median over completed runs on one H200; "–" marks combinations that were not run.

D.4.1 Tuning protocol, search grids and selected configurations

Parameter convention. We tune MARS in (β, γ, η) and VENUS in $(\beta, \gamma, \tau, \eta)$, where η is the peak stepsize and $\tau = 0$ recovers MARS. The implementation converts effective VENUS coordinates to (μ, p, θ). A *configuration* specifies all tuned parameters; a *cell* (η, β, γ) is shared across methods and values of τ. Matched-cell comparisons on the tuning seed are distinct from paired-seed confirmations.

Search and budget. The search proceeded in adaptive *tuning waves*, starting with the comprehensive batch-32 grid and using its results to define focused grids at batch sizes 16 and 64. Table 9 specifies every tested configuration as a union of product blocks. At batch size 32, the initial grids contain 108 configurations per method; extensions add 14 for MARS and 22 for VENUS, giving totals of 122 and 130. Previously evaluated configurations were not repeated. At batch sizes 16 and 64, the methods use identical cells, with 32 and 34 configurations each and VENUS fixed at $\tau = 0.5$.

Selection. We pool the tuning waves at each batch size and select the lowest final validation loss. The general rule was to extend the search outward at boundary optima before confirmation; exceptions for cached variants in empirically flat regions are described in Section D.5.2. All selected exact-method stepsizes and refresh weights lie inside their searched ranges. Ten runs diverged, five per method, all at batch size 32 with $\beta = 0.04$ and $\gamma = 0.13$. Configuration counts include these terminated runs.

D.4.2 Choice of τ

We compared $\tau \in \{0.1, 0.5, 0.75\}$ once, on the tuning seed at batch size 32, then fixed $\tau = 0.5$ at the other batch sizes.

Comparison set. The comprehensive VENUS grid contains 36 cells per τ, all also tested for MARS. Within these cells, MARS diverged in four, all with $\beta = 0.04$ and $\gamma = 0.13$; VENUS diverged in two, three and zero cells at $\tau = 0.1$, 0.5 and 0.75, respectively, all within those four cells. Comparisons across τ therefore use the same 32 completed cells. The $\tau = 0.5$ panel of Figure 14a additionally includes extensions, giving 54 matched cells.

Results. On the 32 common cells, $\tau = 0.5$ gives lower VENUS loss than either alternative in every cell. The mean differences (VENUS minus MARS) are $+0.038$, -0.013 and $+0.010$ for $\tau = 0.1$, 0.5 and 0.75, with VENUS lower in 0, 31 and 2 cells, respectively. The best VENUS losses are 3.5398, 3.4972 and 3.5188, compared with 3.4980 for MARS on these cells.

Including extensions, VENUS at $\tau = 0.5$ has lower loss in $46/54$ cells at batch size 32, with mean difference -0.0085. At batch size 16 it is lower in $32/32$ cells, with mean difference -0.0165; at batch size 64, in $17/34$, with mean difference -0.0001 (Figure 14). These comparisons support the batch-size trend beyond the selected configurations, but the grids differ across batch sizes.

The selected value $\tau = 0.5$ also minimizes the theoretical factor $\chi_\tau = \sqrt{\tau^2 + (1-\tau)^2}$ (Section 3.2.2). This agreement supports a default choice, but does not establish optimality for the full estimator variance or for validation loss beyond the tested configurations. The CNN study uses a finer grid (Section C.7).

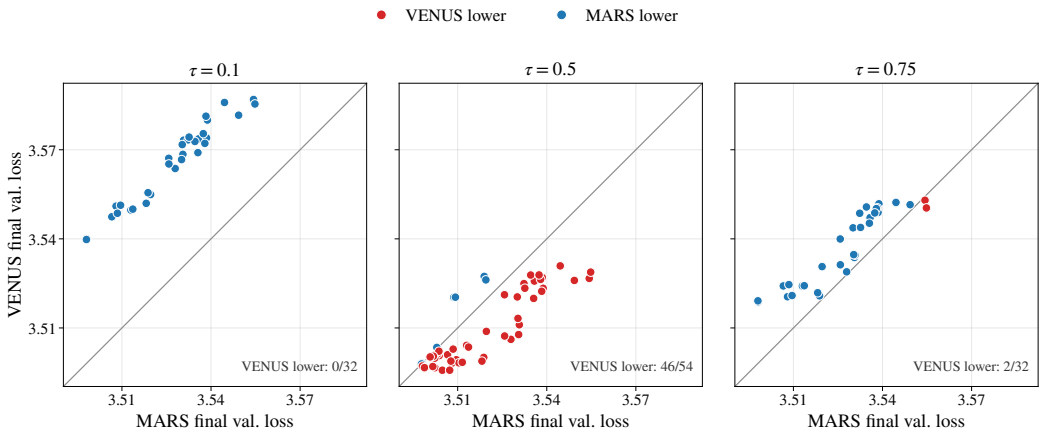
Method	Batch	Swept values and configurations run	#cfg	Best configuration
MARS 2 grads/step	16	$\eta \in \{0.45, 0.9, 1.8, 3.6, 7.2\}$ (5) $\beta \in \{0.01, 0.02, 0.04, 0.08\}$ (4) $\gamma \in \{0.0015, 0.005\}$ (2) run: all (η, β) pairs except $(0.45, 0.08)$, $(3.6, 0.01)$, $(7.2, 0.01)$ and $(7.2, 0.02)$, each at both γ	32	$(\beta, \gamma, \eta)=$ $(0.02, 0.0015, 0.9)$
	32	$\eta \in \{0.9, 1.8, 3.6, 7.2, 14, 28\}$ (6) $\beta \in \{0.01, 0.02, 0.04, 0.06, 0.11, 0.32\}$ (6) $\gamma \in \{0.0005, 0.0015, 0.005, 0.015, 0.045, 0.13\}$ (6) run: $\{0.9, 1.8, 3.6, 7.2, 14, 28\} \times \{0.04, 0.11, 0.32\}$ $\times\{0.0005, 0.0015, 0.005, 0.015, 0.045, 0.13\}$ (108) $\cup\, \{0.9, 1.8\} \times \{0.01, 0.02\} \times \{0.0015, 0.005\}$ (8) $\cup\, \{0.9, 1.8, 3.6\} \times \{0.06\} \times \{0.0015, 0.005\}$ (6)	122	$(\beta, \gamma, \eta)=$ $(0.04, 0.005, 1.8)$
	64	$\eta \in \{1.8, 3.6, 7.2, 14\}$ (4) $\beta \in \{0.04, 0.08, 0.16\}$ (3) $\gamma \in \{0.0005, 0.0015, 0.005, 0.015, 0.045, 0.13\}$ (6) run: $\{1.8, 3.6, 7.2\} \times \{0.04, 0.08, 0.16\}$ $\times\{0.0015, 0.005\}$ (18) $\cup\, \{14\} \times \{0.08, 0.16\} \times \{0.0015, 0.005\}$ (4) $\cup\, \{3.6, 7.2, 14\} \times \{0.08\} \times \{0.0005, 0.045\}$ (6) $\cup\, \beta = 0.08$ at $(\eta, \gamma) \in \{(1.8, 0.045), (1.8, 0.13), (3.6, 0.015), (3.6, 0.13), (7.2, 0.015), (7.2, 0.13)\}$ (6)	34	$(\beta, \gamma, \eta)=$ $(0.08, 0.005, 3.6)$
VENUS 3 grads/step	16	the same (η, β, γ) configurations as MARS $\tau \in \{0.5\}$ (1)	32	$(\beta, \gamma, \tau, \eta)=$ $(0.02, 0.0015, 0.5, 1.8)$
	32	$\eta \in \{0.9, 1.8, 3.6, 7.2, 14\}$ (5) $\beta \in \{0.01, 0.02, 0.04, 0.06, 0.11, 0.32\}$ (6) $\gamma \in \{0.0005, 0.0015, 0.005, 0.015, 0.045, 0.13\}$ (6) $\tau \in \{0.1, 0.5, 0.75\}$ (3) run: $\{1.8, 3.6, 7.2, 14\} \times \{0.04, 0.11, 0.32\}$ $\times\{0.0015, 0.015, 0.13\}$, at each τ (108) and, at $\tau = 0.5$ only, $\cup\, \{0.9, 1.8\} \times \{0.01, 0.02\} \times \{0.0015, 0.005\}$ (8) $\cup\, \{0.9, 1.8, 3.6\} \times \{0.04, 0.06\} \times \{0.0015, 0.005\}$ (12) $\cup\, \{1.8, 3.6\} \times \{0.04\} \times \{0.0005, 0.045\}$ (4); the two configurations already in the main grid are counted once	130	$(\beta, \gamma, \tau, \eta)=$ $(0.06, 0.0015, 0.5, 3.6)$
	64	the same (η, β, γ) configurations as MARS $\tau \in \{0.5\}$ (1)	34	$(\beta, \gamma, \tau, \eta)=$ $(0.08, 0.045, 0.5, 3.6)$

Table 9: Search space and selected single-seed optimum for each method and batch size. For readability each row first gives the values swept on each axis, with the number of values in parentheses, and then the configurations that were actually run: unlike the CNN grids of Table 4, this search is not one full Cartesian product, but a union of the product blocks of the tuning waves (Section D.4.1), and #cfg counts the configurations run on the tuning seed, including diverged ones. MARS configurations are (β, γ, η) and VENUS configurations $(\beta, \gamma, \tau, \eta)$; stepsizes η are in units of 10^{-4}. The best configuration is the one with the lowest seed-52 final validation loss; multi-seed results follow in Sections D.4.3–D.4.5.

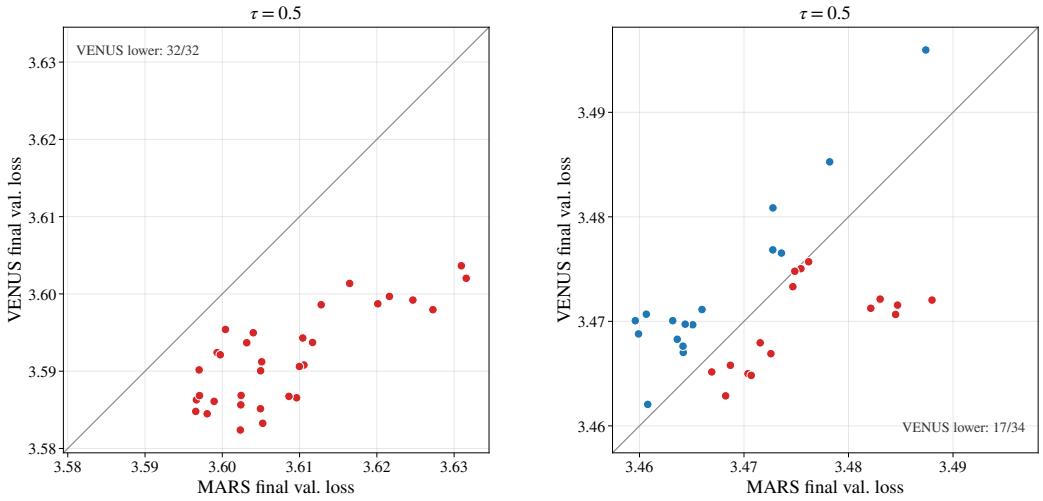
D.4.3 Batch 16

Tuning. No run diverged. The response plots in Figure 15 show that the best tested refresh weight generally increases with the stepsize. When γ changes between 0.0015 and 0.005, the largest observed loss changes are 0.006 for MARS and 0.003 for VENUS across the tested grid. Both methods select $\beta = 0.02$ and $\gamma = 0.0015$; VENUS selects twice the MARS stepsize (Table 9).

Confirmation. VENUS has lower final validation loss on all six seeds (Table 10), with mean difference -0.0109 and 95% interval $[-0.0150, -0.0068]$. The paired t-test gives $p = 0.001$ and the sign test $p = 0.03$.



(a) Batch size 32: VENUS at $\tau \in \{0.1, 0.5, 0.75\}$.



(b) Batch size 16: VENUS at $\tau = 0.5$.

(c) Batch size 64: VENUS at $\tau = 0.5$.

Figure 14: Final validation loss of VENUS against MARS at identical (η, β, γ), on the tuning seed and over all tuning waves. Each point is one cell (η, β, γ) in which both runs completed. Points below the diagonal (red) are cells in which VENUS reaches the lower final validation loss; points above it (blue) are those in which MARS does. At batch size 32 the three values of τ share the 32 cells of the comprehensive grid; the panel for $\tau = 0.5$ has more points because the extension waves were run at that value only.

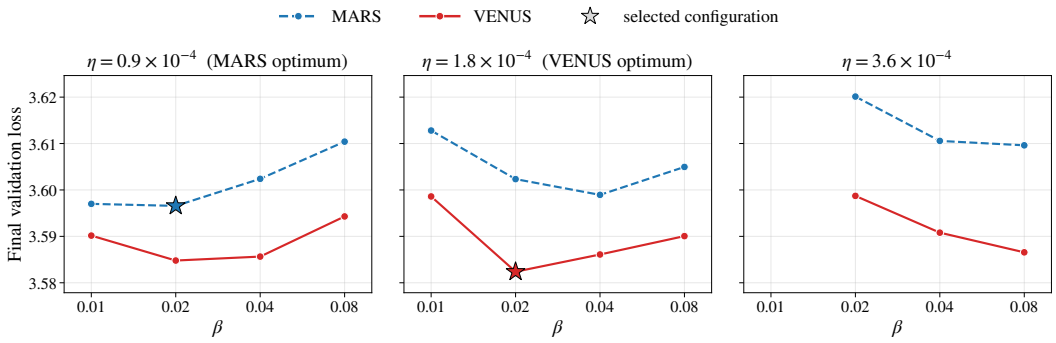
D.4.4 Batch 32

Tuning. Figure 16 shows that the best tested β increases with η for both methods; the advantage of VENUS decreases at the smallest refresh weights in these slices. At $\beta = 0.04$ and $\eta \in \{1.8, 3.6\} \times 10^{-4}$, the largest loss ranges are 0.0038 for MARS and 0.0031 for VENUS over $\gamma \in \{0.0005, 0.0015, 0.005, 0.015, 0.045\}$. All divergent configurations use $\gamma = 0.13$ (Section D.4.1).

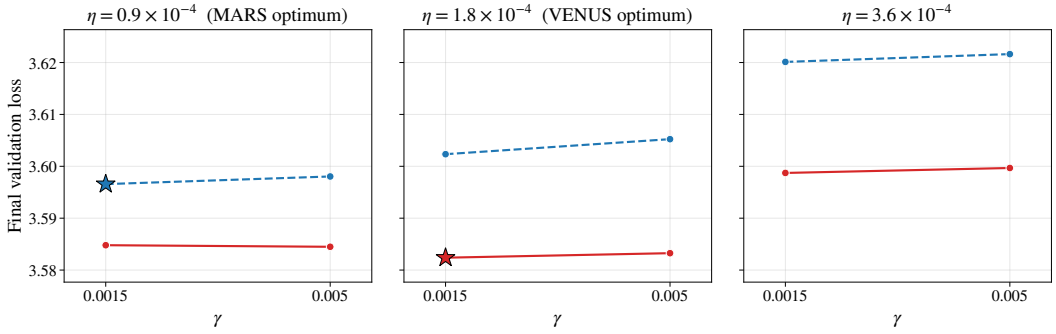
Sensitivity to the stepsize. At $\gamma = 0.0015$, VENUS has a smaller loss range over the shared stepsizes at $\beta = 0.04$ and $\beta = 0.11$, but not at $\beta = 0.06$ (Figure 17). For example, at $\beta = 0.11$ the ranges are 0.0077 for VENUS and 0.0239 for MARS. This observation is limited to the tested slices and tuning seed. Replacing $\gamma = 0.0015$ with the selected MARS value $\gamma = 0.005$ changes the MARS loss by at most 0.0025 on these configurations.

Seed	MARS	VENUS	VENUS − MARS
53	3.5955	3.5840	−0.0115
54	3.5972	3.5874	−0.0098
55	3.5970	3.5835	−0.0136
56	3.5926	3.5824	−0.0102
57	3.5963	3.5919	−0.0044
58	3.5962	3.5802	−0.0160
Mean ± SD	3.5958 ± 0.0017	$\mathbf{3.5849 \pm 0.0042}$	$\mathbf{-0.0109 \pm 0.0039}$

Table 10: Batch size 16, confirmation seeds: final validation loss of the selected configurations, MARS at $(\beta, \gamma, \eta) = (0.02, 0.0015, 0.9)$ and VENUS at $(\beta, \gamma, \tau, \eta) = (0.02, 0.0015, 0.5, 1.8)$, with η in units of 10^{-4}. The last column is the paired difference; it is negative when the VENUS loss is the lower of the two. A lower loss is better.



(a) Final validation loss against β, at $\gamma = 0.0015$.



(b) Final validation loss against γ, at $\beta = 0.02$.

Figure 15: Batch size 16, tuning seed: MARS and VENUS ($\tau = 0.5$) at the stepsize selected for MARS ($\eta = 0.9 \times 10^{-4}$), the stepsize selected for VENUS ($\eta = 1.8 \times 10^{-4}$) and one step above ($\eta = 3.6 \times 10^{-4}$). Stars mark the configuration selected for each method. At $\eta = 3.6 \times 10^{-4}$, $\beta = 0.01$ was not run (Section D.4.1).

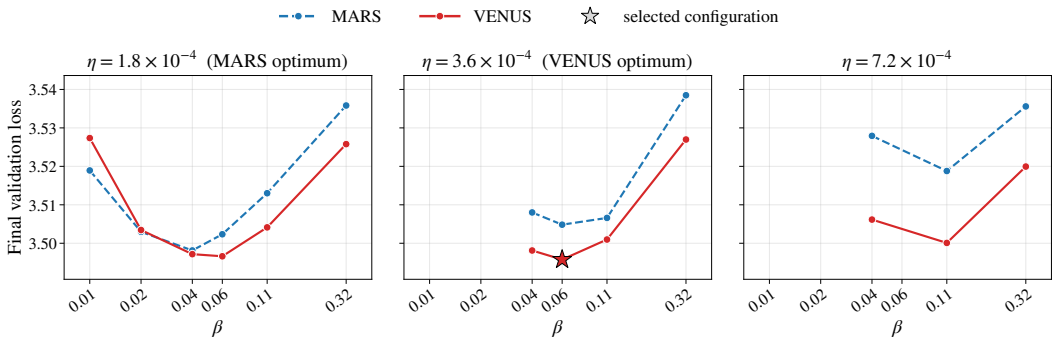
Confirmation. For the selected configurations in Table 9, VENUS has lower loss on five of six seeds (Table 11). The mean difference is -0.0028, with 95% interval $[-0.0063, +0.0006]$; the paired t-test gives $p = 0.09$ and the sign test $p = 0.22$. The difference is therefore inconclusive. A second VENUS configuration, $(\beta, \gamma, \tau, \eta) = (0.04, 0.0015, 0.5, 1.8 \times 10^{-4})$, has mean difference -0.0010 relative to the selected MARS configuration, negative on four of six seeds.

D.4.5 Batch 64

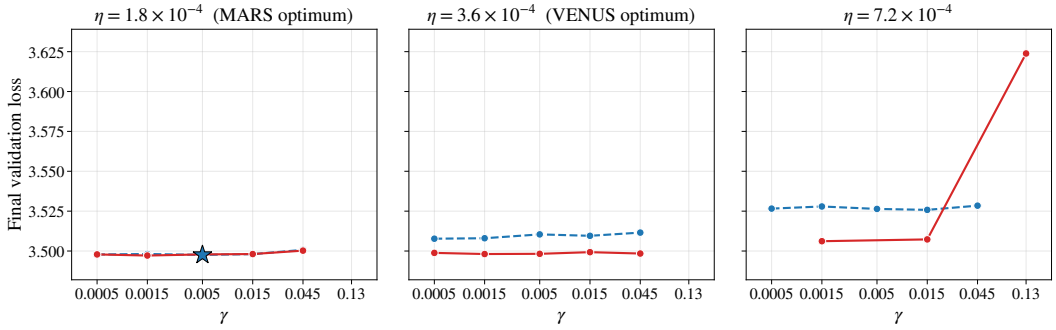
Tuning. No run diverged. Both methods select $\beta = 0.08$ and $\eta = 3.6 \times 10^{-4}$, but VENUS selects a larger correction, $\gamma = 0.045$, than MARS, $\gamma = 0.005$ (Table 9). At this (β, η), loss varies by 0.0086

Seed	MARS	VENUS	VENUS − MARS
53	3.4985	3.4961	−0.0024
54	3.4986	3.4978	−0.0008
55	3.4999	3.4932	−0.0066
56	3.4960	3.4901	−0.0059
57	3.5020	3.5043	+0.0023
58	3.4988	3.4952	−0.0036
Mean ± SD	3.4990 ± 0.0019	$\mathbf{3.4961 \pm 0.0048}$	$\mathbf{-0.0028 \pm 0.0033}$

Table 11: Batch size 32, confirmation seeds: final validation loss of the selected configurations, MARS at $(\beta, \gamma, \eta) = (0.04, 0.005, 1.8)$ and VENUS at $(\beta, \gamma, \tau, \eta) = (0.06, 0.0015, 0.5, 3.6)$, with η in units of 10^{-4}. The last column is the paired difference; it is negative when the VENUS loss is the lower of the two. A lower loss is better.



(a) Final validation loss against β, at $\gamma = 0.0015$.



(b) Final validation loss against γ, at $\beta = 0.04$.

Figure 16: Batch size 32, tuning seed: MARS and VENUS ($\tau = 0.5$) at the stepsize selected for MARS ($\eta = 1.8 \times 10^{-4}$), the stepsize selected for VENUS ($\eta = 3.6 \times 10^{-4}$) and one step above ($\eta = 7.2 \times 10^{-4}$). A star marks a selected configuration, and it appears only in the panel that contains it: the VENUS star appears in the upper panel at $\eta = 3.6 \times 10^{-4}$ and $\beta = 0.06$; the MARS star appears in the lower panel at $\eta = 1.8 \times 10^{-4}$ and $\gamma = 0.005$. Settings that were not run, and the runs that diverged at $\gamma = 0.13$, are missing from the curves.

for VENUS and 0.0012 for MARS over $\gamma \in \{0.0005, 0.0015, 0.005, 0.015, 0.045\}$; $\gamma = 0.13$ gives higher loss for both (Figure 18).

Sensitivity to the stepsize. At $\gamma = 0.005$, VENUS has a smaller loss range in all three displayed β slices, but MARS achieves the lowest loss across them (Figure 19). At $\beta = 0.08$, their ordering reverses between $\eta = 3.6 \times 10^{-4}$ and 7.2×10^{-4}; at $\beta = 0.04$, MARS is lower throughout the tested range, whereas at $\beta = 0.16$, VENUS is lower throughout.

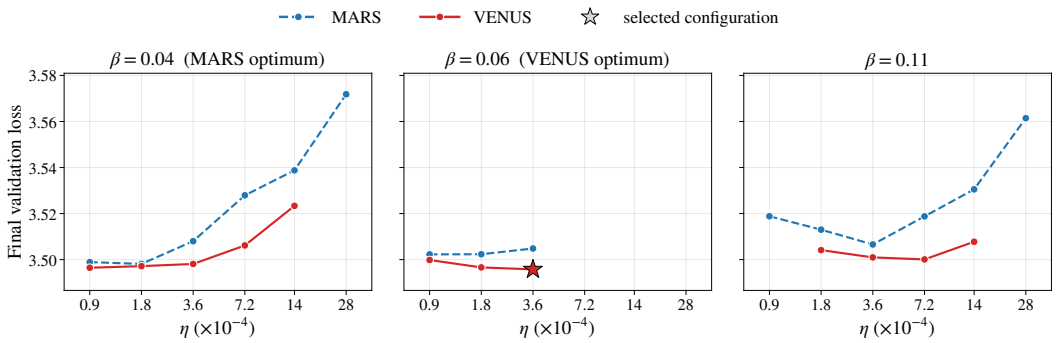


Figure 17: Batch size 32, tuning seed: final validation loss against the peak stepsize η at three values of β, with $\gamma = 0.0015$. Over the shared stepsizes, VENUS ($\tau = 0.5$) has a smaller loss range at $\beta = 0.04$ and $\beta = 0.11$, but not at $\beta = 0.06$. The panels also differ in how many stepsizes were run: MARS covers η up to 28×10^{-4}, and the comparison in the text uses only the stepsizes that both methods ran.

Seed	MARS	VENUS	VENUS − MARS
53	3.4605	3.4653	+0.0047
54	3.4581	3.4659	+0.0078
55	3.4592	3.4621	+0.0029
56	3.4582	3.4634	+0.0053
57	3.4610	3.4709	+0.0099
58	3.4558	3.4599	+0.0041
Mean ± SD	**3.4588 ± 0.0019**	3.4646 ± 0.0038	**+0.0058 ± 0.0026**

Table 12: Batch size 64, confirmation seeds: final validation loss of the selected configurations, MARS at $(\beta, \gamma, \eta) = (0.08, 0.005, 3.6)$ and VENUS at $(\beta, \gamma, \tau, \eta) = (0.08, 0.045, 0.5, 3.6)$, with η in units of 10^{-4}. The last column is the paired difference; it is positive when the MARS loss is the lower of the two. A lower loss is better.

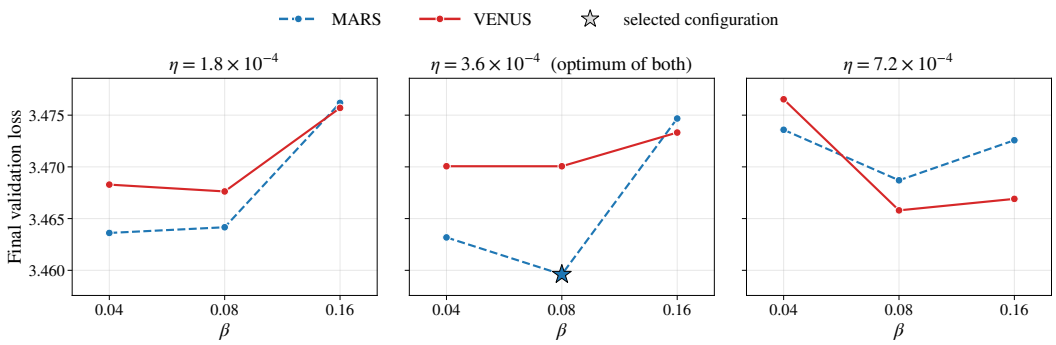
Confirmation. MARS has lower final validation loss on all six seeds (Table 12). The mean difference (VENUS minus MARS) is $+0.0058$, with 95% interval $[+0.0031, +0.0085]$; the paired t-test gives $p = 0.003$ and the sign test $p = 0.03$.

D.4.6 Comparison across batch sizes

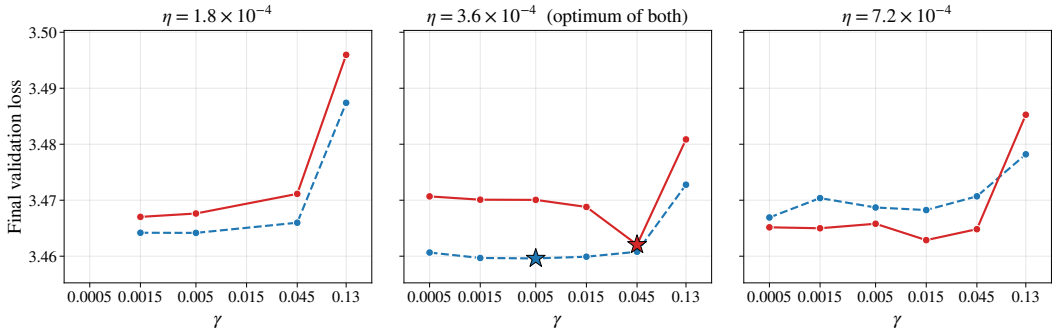
Curve construction. Figures 20 and 21 compare the selected configurations over the six confirmation seeds at matched gradient-token counts. Curves show means and bands show ± 1 sample standard deviation, not confidence intervals. Validation loss is evaluated at the 12 common checkpoints without interpolation or smoothing. Training loss is the pre-update loss on the current ζ minibatch, logged at fixed step intervals of approximately 0.5% of the run budget. Each training-loss curve is linearly interpolated onto 240 equally spaced positions from 2% to 100% of the budget and smoothed with a centred 25-position moving average (approximately 0.277B gradient tokens, or 10.3% of the budget) before averaging over seeds; the window is truncated at the endpoints.

Validation loss during training. The mean difference (VENUS minus MARS) is most favourable to VENUS near 1.12B gradient tokens: -0.0220, -0.0131 and -0.0028 at batch sizes 16, 32 and 64, respectively (Figure 21b). It remains negative at every scheduled checkpoint for batch sizes 16 and 32, but becomes less negative towards the end. At batch size 64, MARS is lower early and late, while VENUS is briefly lower in the middle. The ordering therefore changes during training.

Training loss. For each seed, we average the smoothed training-loss difference over the final quarter of the budget. The resulting six values have mean ± SD of -0.0193 ± 0.0138, -0.0149 ± 0.0249 and -0.0059 ± 0.0249 at batch sizes 16, 32 and 64, respectively. Paired t-tests give $p = 0.02$, 0.20



(a) Final validation loss against β, at $\gamma = 0.005$.



(b) Final validation loss against γ, at $\beta = 0.08$.

Figure 18: Batch size 64, tuning seed: MARS and VENUS ($\tau = 0.5$) at the stepsize selected for both methods ($\eta = 3.6 \times 10^{-4}$) and one step on either side of it. Stars mark the selected configurations, which share $\beta = 0.08$ and $\eta = 3.6 \times 10^{-4}$ and differ in γ, so both appear in the middle panel of the lower figure and only the MARS star appears in the upper one. Settings that were not run are missing from the curves.

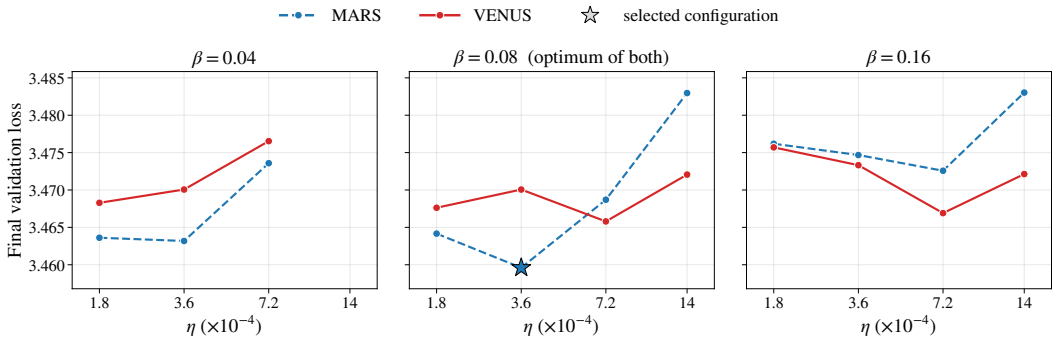
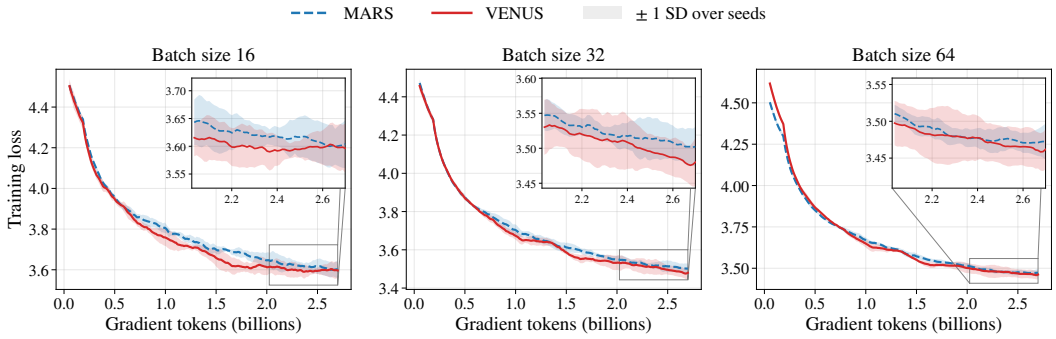
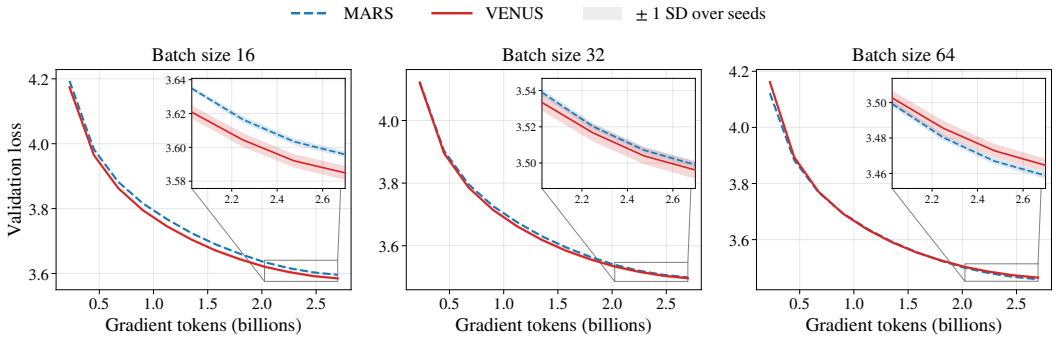


Figure 19: Batch size 64, tuning seed: final validation loss against the peak stepsize η at three values of β, with $\gamma = 0.005$. VENUS ($\tau = 0.5$) has a smaller loss range in each displayed β slice, while MARS achieves the lowest loss across these settings. At $\beta = 0.08$, the curves cross between $\eta = 3.6 \times 10^{-4}$ and $\eta = 7.2 \times 10^{-4}$.

and 0.58, so only the batch-16 difference is significant at the unadjusted 5% level. At batch size 64, the mean training-loss difference over this interval and the final validation-loss difference have opposite signs. The non-significant training-loss differences do not establish equivalence.



(a) Training loss, smoothed as described in the text: a centred moving average over 10.3% of the budget.



(b) Validation loss at the 12 scheduled checkpoints.

Figure 20: The confirmed configurations of MARS and VENUS at the three batch sizes, against gradient tokens. Each method appears at the single configuration selected for it in Sections D.4.3–D.4.5 and listed in Tables 10–12, rerun on the six confirmation seeds 53–58. The curve is the mean over those six runs and the band is ± 1 standard deviation over them: it measures seed-to-seed variability at one fixed configuration, not variation across configurations, and it is the spread of individual runs rather than an interval for the mean. The inset magnifies the last quarter of the budget, where the two methods are close.

D.5 Approximate (cached) estimators

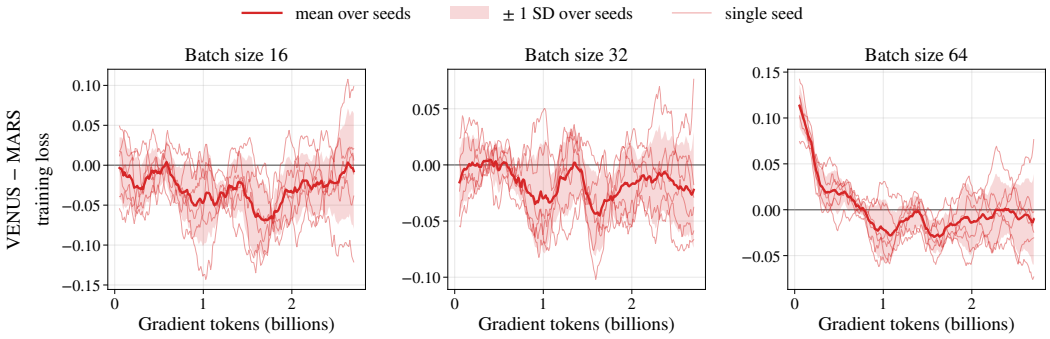
D.5.1 Definition and cost

The cached transport difference is $g_k^+ - g_{k-1}^+$, where $g_k^+ = \nabla f_{\zeta_k}(x_k)$ and g_{k-1}^+ is stored from the previous iteration. Its endpoints use different minibatches, unlike exact same-sample transport; this is the approximation used by MARS (Yuan et al., 2025). Approximate VENUS additionally evaluates an independent gradient on ξ_k at the same iterate x_k (Algorithms 3–4).

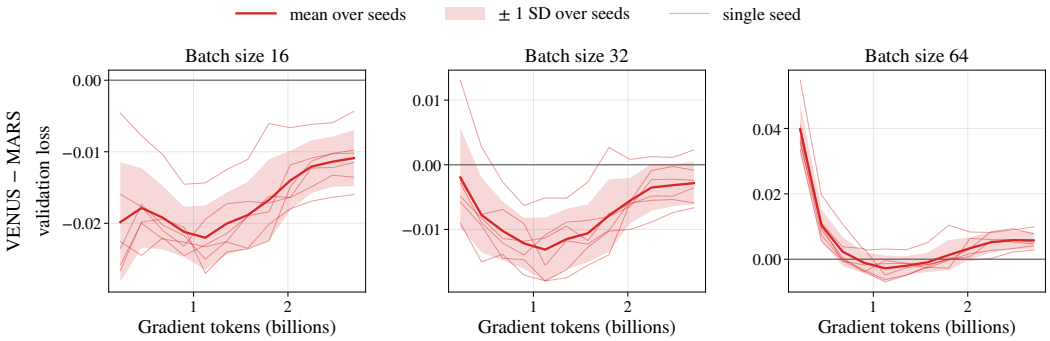
Approximate MARS and VENUS require one and two gradients per step, respectively, permitting 2 and 1.5 times as many updates as their exact counterparts at the same gradient-token budget. Update counts, sampled data and runtimes are reported in Table 8. The convergence analysis of Section 3.2.2 does not cover these cached estimators.

The two pseudocodes are written in the order of Algorithms 1 and 2, so that the only difference from their exact counterparts is the transport pair. Exact MARS and VENUS subtract g_{k+1}^0, taken at x_k on the *current* minibatch ζ_{k+1}; the cached variants subtract g_k^+, which was taken at x_k on the *previous* minibatch ζ_k and is already in memory, so no gradient is evaluated at the previous iterate. Every other line is unchanged, and $\gamma = 0$ again removes the correction.

The implementation differs from this presentation in two ways that leave the recursion intact. It evaluates the current gradients and updates the momentum before taking each step, which is the same sequence read from a different starting point, and it warm-starts the loop rather than being given m_0



(a) Training loss.



(b) Validation loss.

Figure 21: Difference VENUS minus MARS during training, at the three batch sizes, for the same two configurations per batch as in Figure 20. On each seed the difference is taken between the two runs of that seed. Thin lines are the six confirmation seeds, the thick line is their mean and the band is ± 1 sample standard deviation over the seeds. Negative values indicate lower loss for VENUS. The band represents variation across seeds, not a confidence interval for the mean. The vertical scales differ between panels.

Algorithm 3 Approximate (cached) MARS

Require: x_0, m_0, cached g_0^+, horizon K, stepsizes $\{\eta_k\}_{k=0}^{K}$, refresh weight $\beta \in (0, 1]$, correction scale $\gamma \in [0, 1]$

1: **for** $k = 0,1,\ldots,K$ **do**
2: $\quad x_{k+1} \leftarrow x_k + \mathrm{lmo}_{\eta_k}(m_k)$
3: $\quad$ sample one minibatch ζ_{k+1}
4: $\quad g_{k+1}^+ \leftarrow \nabla f_{\zeta_{k+1}}(x_{k+1})$ (g_k^+ is *reused*: one gradient per step)
5: $\quad m_{k+1} \leftarrow (1-\beta)\big(m_k + \gamma(g_{k+1}^+ - g_k^+)\big) + \beta g_{k+1}^+$
6: **end for**

and g_0^+: at the first iteration it sets $m_0 \leftarrow g_0^+$, or $m_0 \leftarrow \tau h_0^+ + (1-\tau)g_0^+$ for the cached VENUS, from gradients evaluated at x_0 and without a transport correction.

D.5.2 Tuning

The cached variants were tuned independently at batch sizes 16 and 32, using the selection protocol of Section D.4.1 and the grids in Table 13. These grids include smaller stepsizes and refresh weights to accommodate the larger number of updates; no run diverged. In some searches, a boundary selection was retained because neighbouring tested settings had similar losses. This includes approximate MARS at batch size 16, with $\eta = 0.225 \times 10^{-4}$, and does not establish optimality outside the tested range.

Algorithm 4 Approximate (cached) VENUS

Require: x_0, m_0, cached g_0^+, horizon K, stepsizes $\{\eta_k\}_{k=0}^K$, refresh weight $\beta \in (0,1)$, correction scale $\gamma \in [0,1]$, sample-splitting weight $\tau \in [0,1]$

for $k = 0,1,\ldots,K$ **do**
  $x_{k+1} \leftarrow x_k + \mathrm{lmo}_{\eta_k}(m_k)$
  sample two fresh, conditionally independent minibatches ξ_{k+1} and ζ_{k+1}
  $h_{k+1}^+ \leftarrow \nabla f_{\xi_{k+1}}(x_{k+1})$
  $g_{k+1}^+ \leftarrow \nabla f_{\zeta_{k+1}}(x_{k+1})$ (g_k^+ is *reused*: two gradients per step)
  $m_{k+1} \leftarrow (1-\beta)\big(m_k + \gamma(g_{k+1}^+ - g_k^+)\big) + \beta\big(\tau h_{k+1}^+ + (1-\tau)g_{k+1}^+\big)$
end for

Method	Batch	Swept values and configurations run	#cfg	Best configuration
approximate MARS 1 grad/step	16	$\eta \in \{0.225, 0.45, 0.9, 1.8\}$ (4) $\beta \in \{0.005, 0.01, 0.02, 0.04\}$ (4) $\gamma \in \{0.0015, 0.005\}$ (2) run: the full product of the three	32	$(\beta,\gamma,\eta)=$ $(0.01, 0.0015, 0.225)$
	32	$\eta \in \{0.225, 0.45, 0.9, 1.8, 3.6\}$ (5) $\beta \in \{0.01, 0.02, 0.04\}$ (3) $\gamma \in \{0.0015, 0.005\}$ (2) run: $\{0.225, 0.45, 0.9, 1.8\} \times \{0.01, 0.02, 0.04\}$ $\times\{0.0015, 0.005\}$ (24) $\cup\, \{3.6\} \times \{0.02, 0.04\} \times \{0.0015\}$ (2)	26	$(\beta,\gamma,\eta)=$ $(0.02, 0.0015, 1.8)$
approximate VENUS 2 grads/step	16	$\eta \in \{0.45, 0.9, 1.8, 3.6\}$ (4) $\beta \in \{0.005, 0.01, 0.02, 0.04\}$ (4) $\gamma \in \{0.0015\}$ (1), $\tau \in \{0.5\}$ (1) run: the full product of the four	16	$(\beta,\gamma,\tau,\eta)=$ $(0.01, 0.0015, 0.5, 0.9)$
	32	$\eta \in \{0.45, 0.9, 1.8, 3.6, 7.2\}$ (5) $\beta \in \{0.01, 0.02, 0.04, 0.08\}$ (4) $\gamma \in \{0.0015, 0.005\}$ (2), $\tau \in \{0.5\}$ (1) run: $\{0.45, 0.9, 1.8, 3.6\} \times \{0.01, 0.02, 0.04\}$ $\times\{0.0015, 0.005\}$ (24) $\cup\, \{1.8, 3.6, 7.2\} \times \{0.04, 0.08\} \times \{0.005\}$, the two configurations already in the first block counted once (4)	28	$(\beta,\gamma,\tau,\eta)=$ $(0.04, 0.005, 0.5, 1.8)$

Table 13: Search space and selected single-seed optimum of the cached variants. The layout follows Table 9: each row first gives the values swept on each axis, with the number of values in parentheses, and then the configurations actually run. Approximate MARS configurations are (β,γ,η) and approximate VENUS configurations (β,γ,τ,η); stepsizes η are in units of 10^{-4}. #cfg counts the configurations run on the tuning seed 52; none diverged.

The tuning budgets are unequal: approximate MARS/VENUS use $32/16$ configurations at batch size 16 and $26/28$ at batch size 32. Exact methods use at least as many configurations as their cached counterparts, so the cached variants' lower losses cannot be attributed to a larger number of tuning runs. Differences in search spaces nevertheless limit these comparisons.

D.5.3 Results

Tables 14 and 15 report all six confirmation seeds and pairwise differences. At both batch sizes, the mean final loss is lowest for approximate MARS, followed by approximate VENUS, exact VENUS and exact MARS. Both cached variants outperform both exact methods on every seed. Approximate MARS also outperforms approximate VENUS on every seed, by 0.0208 on average at batch size 16 and 0.0263 at batch size 32. All pairwise intervals exclude zero except the exact batch-32 comparison.

Approximate VENUS and exact MARS have the same update count and two gradient evaluations per step, but different sampled-data exposure. Approximate VENUS achieves lower mean final loss by 0.0620 and 0.0599 at batch sizes 16 and 32, respectively. Figures 22 and 23 show the trajectories: on

Batch	Seed	MARS	VENUS	approx. MARS	approx. VENUS
16	53	3.5955	3.5840	3.5160	3.5347
	54	3.5972	3.5874	3.5145	3.5390
	55	3.5970	3.5835	3.5128	3.5307
	56	3.5926	3.5824	3.5118	3.5327
	57	3.5963	3.5919	3.5086	3.5355
	58	3.5962	3.5802	3.5139	3.5299
	Mean ± SD	3.5958 ± 0.0017	3.5849 ± 0.0042	$\mathbf{3.5129 \pm 0.0026}$	3.5338 ± 0.0034
32	53	3.4985	3.4961	3.4136	3.4386
	54	3.4986	3.4978	3.4119	3.4420
	55	3.4999	3.4932	3.4150	3.4375
	56	3.4960	3.4901	3.4089	3.4349
	57	3.5020	3.5043	3.4139	3.4445
	58	3.4988	3.4952	3.4128	3.4367
	Mean ± SD	3.4990 ± 0.0019	3.4961 ± 0.0048	$\mathbf{3.4127 \pm 0.0022}$	3.4390 ± 0.0036

Table 14: Final validation loss of the four methods on the six confirmation seeds, at batch sizes 16 and 32. Each method runs at the configuration selected for it, from Tables 9 and 13. Lower is better; the best mean in each block is in bold.

Batch	Difference	Mean ± SD	95% interval	First lower
16	VENUS − MARS	-0.0109 ± 0.0039	$[-0.0150, -0.0068]$	6/6
	approx. MARS − MARS	-0.0829 ± 0.0029	$[-0.0859, -0.0799]$	6/6
	approx. VENUS − MARS	-0.0620 ± 0.0035	$[-0.0657, -0.0584]$	6/6
	approx. MARS − VENUS	-0.0720 ± 0.0060	$[-0.0783, -0.0656]$	6/6
	approx. VENUS − VENUS	-0.0511 ± 0.0030	$[-0.0543, -0.0480]$	6/6
	approx. VENUS − approx. MARS	$+0.0208 \pm 0.0042$	$[+0.0164, +0.0252]$	0/6
32	VENUS − MARS	-0.0028 ± 0.0033	$[-0.0063, +0.0006]$	5/6
	approx. MARS − MARS	-0.0863 ± 0.0013	$[-0.0877, -0.0849]$	6/6
	approx. VENUS − MARS	-0.0599 ± 0.0024	$[-0.0625, -0.0574]$	6/6
	approx. MARS − VENUS	-0.0834 ± 0.0042	$[-0.0879, -0.0790]$	6/6
	approx. VENUS − VENUS	-0.0571 ± 0.0018	$[-0.0590, -0.0552]$	6/6
	approx. VENUS − approx. MARS	$+0.0263 \pm 0.0033$	$[+0.0228, +0.0299]$	0/6

Table 15: The six pairwise differences of Table 14, taken seed by seed. A negative mean means the first method of the pair ends lower. The interval is the 95% Student-t interval for the mean (5 degrees of freedom), and the last column counts the seeds on which the first method is lower. All confidence intervals exclude zero except VENUS against MARS at batch size 32.

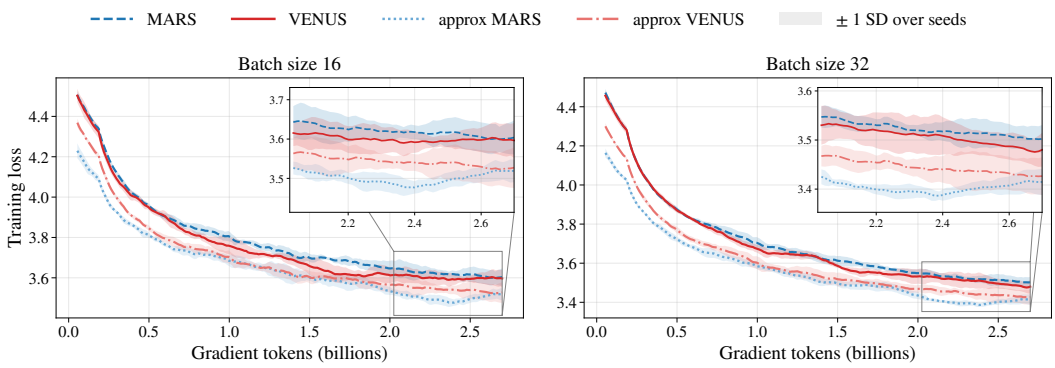
all six seeds at every scheduled checkpoint, approximate VENUS has lower validation loss than both exact methods and higher loss than approximate MARS.

D.5.4 Interpretation and scope

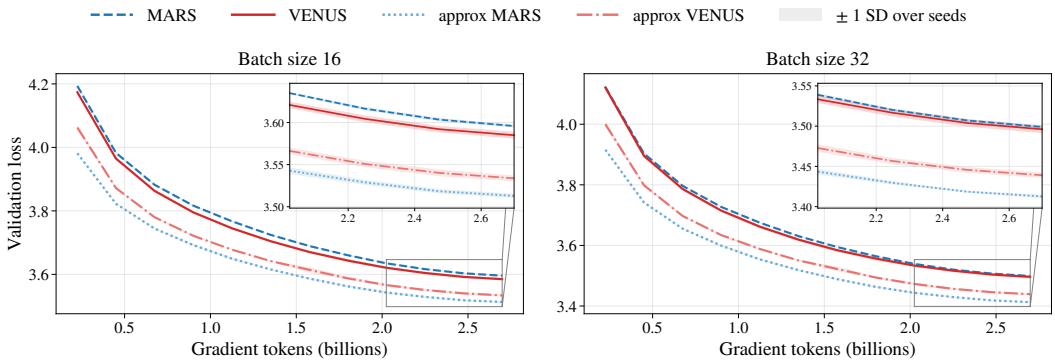
Caching changes the estimator, number of updates and sampled-data exposure simultaneously. The observed gains therefore do not identify the contribution of any one factor. They also depend on the budget definition: equal gradient-token work does not imply equal wall-clock time (Table 8). The unequal cached tuning grids and fixed $\tau = 0.5$ further limit conclusions about the relative performance of the two cached methods.

D.6 Summary of findings

Under the matched gradient-token budget, exact VENUS improves final validation loss at batch size 16, whereas exact MARS performs better at batch size 64; the batch-32 difference remains inconclusive. Among the three sample-splitting weights tested at batch size 32, $\tau = 0.5$ performs best and is used at the other batch sizes. Both cached variants outperform their exact counterparts at batch sizes 16 and 32, with approximate MARS achieving the lowest loss. The exact-method batch-size



(a) Training loss, smoothed as in Section D.4.6.



(b) Validation loss at the 12 scheduled checkpoints.

Figure 22: The four methods at their selected configurations, on the six confirmation seeds, against gradient tokens. Colour marks the family, blue for MARS and red for VENUS, and the line style marks the variant, so that exact and cached versions of a method can be read together. Each curve is the mean over the six seeds and the band is ± 1 standard deviation over them. The inset magnifies the last quarter of the budget. The cached variants were run at batch sizes 16 and 32 only.

pattern is consistent with a benefit from reduced sampling noise, but the experiments do not isolate that mechanism.

D.7 Limitations

- *Limited settings.* The study uses one 124M-parameter model, FineWeb, one gradient-token budget and one schedule. Exact methods are tested at three batch sizes and cached methods at two; this does not locate a precise batch-size crossover or establish scaling to other settings.
- *Selection and uncertainty.* Six fresh seeds confirm configurations selected on one tuning seed. The intervals describe variability at those fixed configurations, not uncertainty from repeating the search. Only batch size 32 includes a second confirmed VENUS configuration.
- *Budget definition.* Gradient-token matching does not equalize sampled data or wall-clock time (Section D.3); the study does not separate their effects from those of the estimator.
- *Tuning scope.* Search budgets differ for the exact batch-32 comparison and both cached comparisons. The τ sweep is restricted to three values at batch size 32 for exact VENUS; $\tau = 0.5$ is fixed elsewhere, including for approximate VENUS.
- *Optimizer scope.* All methods use the same spectral LMO geometry for the Transformer blocks and AdamW for the tied embedding. No standalone AdamW or Muon baseline is evaluated.

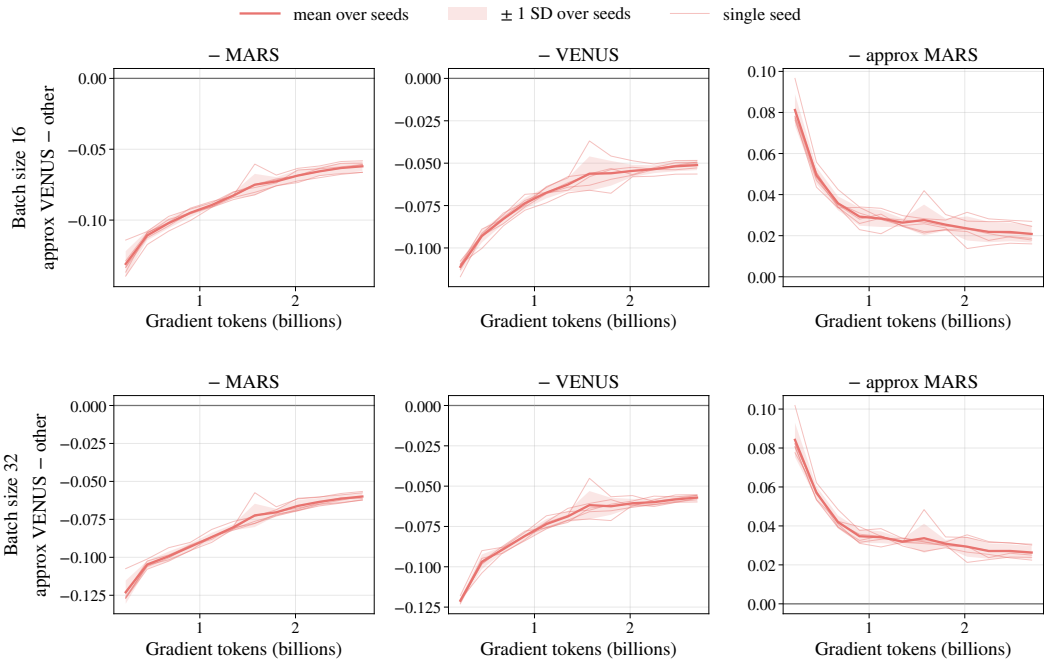


Figure 23: Approximate VENUS against each of the other three methods during training: the validation-loss difference taken seed by seed, at batch size 16 (top) and 32 (bottom). Thin lines are the six confirmation seeds, the thick line is their mean and the band is ± 1 standard deviation over the seeds. Negative values indicate lower loss for approximate VENUS. On every seed and checkpoint, its validation loss is lower than that of either exact method and higher than that of approximate MARS. The vertical scales differ between panels.

E COVARIANCE SIGN-FLIP CONSTRUCTION

The exact comparison in Appendix B shows that sample coupling need not always reduce, or always increase, the innovation variance. This appendix makes that statement concrete through a deliberately simple two-component construction. Its purpose is a constructive existence result: the refresh–transport covariance can change the variance-optimal amount of sample decoupling. It is not an empirical study of full MARS or VENUS optimization trajectories.

E.1 EXACT COVARIANCE MECHANISM

We derive the variance difference in Eq. (13) and the optimal split discussed in Section 6, retaining iteration subscripts to make the conditioning explicit.

Condition on the optimization history through step k, and write $\mathbb{E}_k[\cdot]$ for conditional expectation. Let ζ and ξ be conditionally independent draws from the same component-oracle distribution. At a fixed pair (x_k,x_{k+1}), define the centered fresh-gradient noises

$$z_{k+1}^{\zeta} := \nabla f_{\zeta}(x_{k+1}) - \nabla f(x_{k+1}), \qquad z_{k+1}^{\xi} := \nabla f_{\xi}(x_{k+1}) - \nabla f(x_{k+1}). \tag{65}$$

The raw same-sample transport difference and its centered, scaled noise are

$$\Delta_{k+1}^{\zeta} := \nabla f_{\zeta}(x_{k+1}) - \nabla f_{\zeta}(x_k), \tag{66}$$

$$\widetilde{r}_{k+1}^{\zeta} := \gamma\big(\Delta_{k+1}^{\zeta} - \mathbb{E}_k[\Delta_{k+1}^{\zeta}]\big). \tag{67}$$

We abbreviate the relevant conditional second moments as

$$V_{k+1} := \mathbb{E}_k\|z_{k+1}^{\zeta}\|_2^2, \qquad T_{k+1} := \mathbb{E}_k\|\widetilde{r}_{k+1}^{\zeta}\|_2^2, \qquad C_{k+1} := \mathbb{E}_k\left\langle \widetilde{r}_{k+1}^{\zeta}, z_{k+1}^{\zeta} \right\rangle. \tag{68}$$

Because ξ and ζ have the same conditional law, $\mathbb{E}_k\|z_{k+1}^{\xi}\|_2^2 = V_{k+1}$; because they are independent and centered, every cross moment involving z_{k+1}^{ξ} and a centered ζ-measurable quantity vanishes.

At the same state and effective parameters (β,γ,τ), the centered innovations are

$$q_{k+1}^{\mathrm{MARS}} = (1-\beta)\widetilde{r}_{k+1}^{\zeta} + \beta z_{k+1}^{\zeta}, \tag{69}$$

$$q_{k+1}^{\mathrm{VENUS}} = (1-\beta)\widetilde{r}_{k+1}^{\zeta} + \beta(1-\tau) z_{k+1}^{\zeta} + \beta\tau z_{k+1}^{\xi}. \tag{70}$$

Expanding the squared norms gives

$$\mathbb{E}_k\|q_{k+1}^{\mathrm{MARS}}\|_2^2 = (1-\beta)^2 T_{k+1} + \beta^2 V_{k+1} + 2\beta(1-\beta)C_{k+1}, \tag{71}$$

$$\mathbb{E}_k\|q_{k+1}^{\mathrm{VENUS}}\|_2^2 = (1-\beta)^2 T_{k+1} + \beta^2\big[(1-\tau)^2+\tau^2\big]V_{k+1} + 2\beta(1-\beta)(1-\tau)C_{k+1}. \tag{72}$$

Only the $(1-\tau)z_{k+1}^{\zeta}$ fraction of the VENUS refresh shares its sample with the transport noise. The two covariance contributions are therefore

$$\underbrace{2\beta(1-\beta)C_{k+1}}_{\text{MARS}} \qquad \text{and} \qquad \underbrace{2\beta(1-\beta)(1-\tau)C_{k+1}}_{\text{VENUS}}. \tag{73}$$

Decoupling changes this contribution by $2\beta(1-\beta)\tau C_{k+1}$. Separately, splitting the refresh across the independent ζ and ξ samples produces the nonnegative difference $2\beta^2\tau(1-\tau)V_{k+1}$. Hence the full second-moment difference can be written with both effects visible:

$$\begin{aligned} D_{k+1}(\tau) &:= \mathbb{E}_k\|q_{k+1}^{\mathrm{MARS}}\|_2^2 - \mathbb{E}_k\|q_{k+1}^{\mathrm{VENUS}}\|_2^2 \\ &= \underbrace{2\beta^2\tau(1-\tau)V_{k+1}}_{\text{independent-refresh reduction}} + \underbrace{2\beta(1-\beta)\tau C_{k+1}}_{\text{change in covariance contribution}} \\ &= 2\beta\tau\left[\beta(1-\tau)V_{k+1} + (1-\beta)C_{k+1}\right]. \end{aligned} \tag{74}$$

Positive D_{k+1} means that decoupling lowers the innovation second moment. For $C_{k+1} > 0$ and $\tau > 0$, both displayed terms are positive: decoupling both averages independent refresh noise and removes a fraction τ of the harmful positive covariance. For $C_{k+1} < 0$, the second term is negative: decoupling removes a fraction τ of the beneficial same-sample cancellation. Coupling is preferred only when this loss exceeds the refresh-splitting gain, namely when $(1-\beta)|C_{k+1}| > \beta(1-\tau)V_{k+1}$.

More generally, since the VENUS family contains the coupled endpoint $\tau = 0$, it is natural to ask which amount of decoupling minimizes the innovation variance. For $V_{k+1} > 0$, this is equivalent to maximizing $D_{k+1}(\tau)$ over $[0,1]$, which gives

$$\tau^*_{\mathrm{var}} = \Pi_{[0,1]}\left(\frac{1}{2} + \frac{(1-\beta)C_{k+1}}{2\beta V_{k+1}}\right), \tag{75}$$

where $\Pi_{[0,1]}$ denotes projection onto $[0,1]$. Hence positive covariance moves the optimum above an equal split, while negative covariance moves it toward the coupled endpoint:

$$C_{k+1} > 0 \implies \tau^*_{\mathrm{var}} > \tfrac{1}{2}, \qquad C_{k+1} < 0 \implies \tau^*_{\mathrm{var}} < \tfrac{1}{2}, \tag{76}$$

with the inequalities interpreted after clipping at the endpoints.

The sign of C_{k+1} also has a direct cross-state interpretation. Define $z^\zeta_k := \nabla f_\zeta(x_k) - \nabla f(x_k)$ using the same sample as at x_{k+1}. Then

$$\widetilde{r}^\zeta_{k+1} = \gamma(z^\zeta_{k+1} - z^\zeta_k), \qquad C_{k+1} = \gamma\left(V_{k+1} - \mathbb{E}_k\langle z^\zeta_k, z^\zeta_{k+1}\rangle\right). \tag{77}$$

Consequently, negative covariance requires the same-sample noise at the two states to remain strongly correlated while decreasing in second moment. In fact, for $\gamma > 0$ and $V_{k+1} > 0$, negative covariance first gives $\mathbb{E}_k\langle z^\zeta_k, z^\zeta_{k+1}\rangle > V_{k+1}$. Cauchy–Schwarz then yields

$$\begin{aligned} V_{k+1} &< \mathbb{E}_k\langle z^\zeta_k, z^\zeta_{k+1}\rangle \\ &\le \sqrt{\mathbb{E}_k\|z^\zeta_k\|_2^2\,\mathbb{E}_k\|z^\zeta_{k+1}\|_2^2} = \sqrt{V_k V_{k+1}}, \end{aligned} \qquad \implies \qquad V_k > V_{k+1}, \tag{78}$$

where $V_k := \mathbb{E}_k\|z^\zeta_k\|_2^2$. Moreover, $|C_{k+1}| \le \sqrt{T_{k+1}V_{k+1}}$, so the covariance contribution becomes small whenever the stochastic transport difference is small.

E.2 Two-component sign-flip construction

Let $s \in \{-1, +1\}$ be uniform and, for a sign parameter $c \in \{-1, +1\}$, define

$$f^{(c)}_s(x) = \frac{1}{2}(1+\lambda s)x^2 + c\kappa s x, \qquad \nabla f^{(c)}_s(x) = (1+\lambda s)x + c\kappa s, \tag{79}$$

where $\lambda, \kappa > 0$. Since $\mathbb{E}[s] = 0$, both signs induce exactly the same population problem,

$$f(x) = \mathbb{E}_s[f^{(c)}_s(x)] = \frac{1}{2}x^2, \qquad \nabla f(x) = x. \tag{80}$$

Freeze $x_k = -h$ and $x_{k+1} = 0$, with $h > 0$. At the new point,

$$z^\zeta_{k+1} = c\kappa s_\zeta, \qquad z^\xi_{k+1} = c\kappa s_\xi, \qquad V_{k+1} = \kappa^2. \tag{81}$$

The same-sample gradient difference is

$$\Delta^\zeta_{k+1} = \nabla f^{(c)}_{s_\zeta}(0) - \nabla f^{(c)}_{s_\zeta}(-h) = (1+\lambda s_\zeta)h. \tag{82}$$

The additive $c\kappa s_\zeta$ term cancels from this difference. Therefore

$$\widetilde{r}^\zeta_{k+1} = \gamma\lambda h s_\zeta, \qquad T_{k+1} = \gamma^2\lambda^2 h^2, \qquad C_{k+1} = c\gamma\lambda h\kappa. \tag{83}$$

Thus changing c preserves the population objective, the fixed state, V_{k+1}, and T_{k+1} while reversing C_{k+1}. This is an extremal construction: $|C_{k+1}| = \sqrt{T_{k+1}V_{k+1}}$, so the refresh and transport noises have correlation $c \in \{-1, +1\}$ rather than an intermediate value.

Substitution into Eq. (74) yields

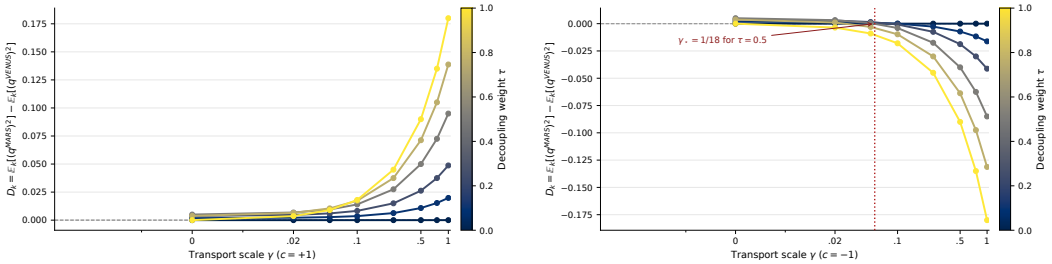
$$D^{(c)}_{k+1}(\tau) = 2\beta\tau\left[\beta(1-\tau)\kappa^2 + (1-\beta)c\gamma\lambda h\kappa\right]. \tag{84}$$

In the negative-sign construction, nonzero decoupling has the larger innovation variance precisely when

$$\gamma > \gamma_\star(\tau) := \frac{\beta(1-\tau)\kappa}{(1-\beta)\lambda h}. \tag{85}$$

c	$(V_{k+1},T_{k+1},C_{k+1})$	S_{k+1}	D_{k+1}	$\mathbb{E}_k\|q^{\mathrm{MARS}}\|_2^2$	$\mathbb{E}_k\|q^{\mathrm{VENUS}}\|_2^2$	τ^*_{var}
$+1$	$(1,1,+1)$	$+0.95$	$+0.095$	1.000	0.905	1
-1	$(1,1,-1)$	-0.85	-0.085	0.640	0.725	0

Table 16: Exact baseline moments. The tuple in the second column is $(V_{k+1},T_{k+1},C_{k+1})$; $S_{k+1} = \beta(1-\tau)V_{k+1} + (1-\beta)C_{k+1}$ is evaluated at $\tau = 0.5$.



(a) Positive-sign construction ($c = +1$). (b) Negative-sign construction ($c = -1$).

Figure 24: Exact sensitivity of the innovation-second-moment difference D_{k+1} to the decoupling weight τ and transport scale γ. Positive values favor decoupling. The dotted marker in the negative-sign panel identifies $\gamma_\star = 1/18$ at $\tau = 0.5$.

For the baseline

$$\beta = 0.1, \qquad \tau = 0.5, \qquad \gamma = \lambda = h = \kappa = 1, \tag{86}$$

the exact values are summarized in Table 16. The threshold in Eq. (85) is $\gamma_\star = 1/18$. At these deliberately extreme parameters, the variance-optimal choices lie at opposite endpoints of the decoupling interval.

Figure 24 shows the exact dependence on τ and γ. At $\tau = 0$, the estimators coincide. At $\gamma = 0$, both constructions have $C_{k+1} = 0$ and an interior split reduces refresh variance. For $\gamma > 0$, stronger transport amplifies harmful same-sample reinforcement when $c = +1$ and beneficial cancellation when $c = -1$; in the latter case the preference reverses only after crossing Eq. (85).

A limited Euclidean interpretation. For this quadratic construction, the variance ordering has an immediate local interpretation under the deliberately simplified update $x^+ = x - \eta q$. At the population minimizer $x = 0$,

$$\mathbb{E}\big[f(x^+_{\mathrm{MARS}}) - f(x^+_{\mathrm{VENUS}})\big] = \frac{\eta^2}{2} D_{k+1}. \tag{87}$$

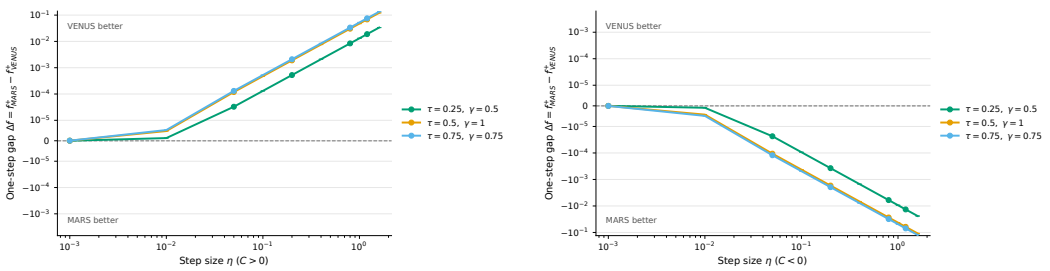
This is an algebraic consequence of the quadratic construction, not independent optimization evidence. It is also not an LMO statement: a one-dimensional norm-ball LMO depends on the sign rather than the magnitude of its input, and the full algorithms update the recursive estimator m_k, not the isolated innovation q_{k+1}.

Figure 25 visualizes this exact local relation for three (τ,γ) choices. The sign reversal mirrors D_{k+1} and the magnitude grows quadratically with η, exactly as Eq. (87) requires. The figure is therefore an illustration of the Euclidean corollary, not an additional optimization experiment.

E.3 Scope and Limitations

The construction isolates an exact fixed-state mechanism, but it is intentionally extreme. It is one-dimensional, has two Rademacher components, and attains perfect correlation ± 1. Although the two signs have identical population objectives and identical new-point refresh and transport second moments, they do not have identical old-point noise. Indeed,

$$z_k^\zeta = s_\zeta(c\kappa - \lambda h), \tag{88}$$



(a) Positive-sign construction ($c = +1$). (b) Negative-sign construction ($c = -1$).

Figure 25: Exact one-step Euclidean objective gap $\mathbb{E}[f(x^+_{\mathrm{MARS}}) - f(x^+_{\mathrm{VENUS}})]$ at the population minimizer. Positive values favor decoupling and negative values favor coupling. The curves visualize the algebraic relation $\eta^2 D_{k+1}/2$; they do not represent recursive MARS or VENUS trajectories.

so under the baseline parameters the $c = +1$ old-point oracle is noiseless, whereas the $c = -1$ old-point noise has variance 4. Moreover, $x_{k+1} = 0$ is the population minimizer, and at $\lambda = 1$ one component of Eq. (79) is linear. These choices make the sign-flip algebra transparent; they are not intended as a realistic model of neural-network stochastic gradients.

The result is also local and Euclidean. It does not evolve the estimator memory, apply the spectral LMO, or preserve a prescribed covariance as an iterate moves. At a fixed state, ordinary minibatch averaging scales V_{k+1} and C_{k+1} together, so it does not by itself change the sign of the score in Eq. (74). Consequently, the sign-flip construction alone does not explain the batch-size crossover observed in the neural-network experiments. Appendix H reports plug-in measurements of V_{k+1}, C_{k+1}, and $D_{k+1}(\tau)$ along three actual Transformer trajectories. Establishing a causal connection between these measurements and the batch-size crossover remains open.

The supported conclusion is therefore deliberately narrow: refresh–transport covariance can change the variance-optimal amount of sample decoupling. Positive covariance moves the optimum toward stronger decoupling; sufficiently negative covariance moves it toward coupling through same-sample cancellation.

F When does sample decoupling reduce variance on CIFAR-10?

Summary. We measure the refresh–transport covariance C_{k+1} along CIFAR-10 training runs to see whether it ever makes coupling preferable. Negative covariance is rare at the state level: it occurs in 2.87% of measured states, although 23.9% of trajectories contain at least one such state. It never erases the variance reduction from averaging independent refresh samples. Positive covariance can be much larger and can favor stronger decoupling, so the variance-optimal split varies from state to state.

F.1 Experimental protocol

We use the CifarNet architecture and the 45,000-image CIFAR-10 training split described in Appendix C.1. BatchNorm makes a minibatch, rather than an individual image, the stochastic oracle component. For batch size b, oracle seed o fixes one augmentation and permutation of the training images and partitions them into

$$M_b = \left\lfloor \frac{45{,}000}{b} \right\rfloor \quad \text{components}, \qquad f_{o,b}(x) = \frac{1}{M_b} \sum_{j=1}^{M_b} \ell_{o,b,j}(x). \tag{89}$$

Each $\ell_{o,b,j}$ is the label-smoothed cross-entropy used in training, with label-smoothing parameter 0.2. The unused remainder contains fewer than b images. All losses and moments in this appendix refer to this batch-indexed finite-oracle objective, not to the CIFAR-10 data distribution. During training, the component indices ζ and ξ are sampled independently with replacement, so they may coincide. The augmented images stay fixed throughout a run.

Quantity	Covariance map $\tau_{\text{train}} \in \{0,.5\}$
Batch b	$\{16,32,64,128\}$
β,γ,η	$\beta \in \{.02,.1\}$, $\gamma = 1$, $\eta = .02$
Training	11,250 updates; three gradient calls per update
Seeds	20 training seeds for oracle 1234; 7 for each of 1235, 1236
Diagnostics	10 fixed progress points from 0.2% to 99%
Moments	exact enumeration of all M_b components; global Euclidean norm

Table 17: Protocol for the 544-trajectory covariance map. Unlike Algorithm 1, the coupled training arm retains the unused independent ξ-gradient evaluation (with $\tau_{\text{train}} = 0$) to match both gradient-call and update counts: three calls per update for each arm.

F.2 Mechanism and notation

At consecutive states (x_k,x_{k+1}), define the component-gradient noise

$$z_j^\zeta := \nabla f_\zeta(x_j) - \nabla f(x_j), \qquad j \in \{k,k+1\}. \tag{90}$$

We call z_{k+1}^ζ the *fresh-gradient noise*; its second moment is $V_{k+1} := \mathbb{E}_k\|z_{k+1}^\zeta\|_2^2$. The centered *transport noise* is

$$\widetilde{r}_{k+1}^\zeta := \gamma\big(z_{k+1}^\zeta - z_k^\zeta\big), \qquad T_{k+1} := \mathbb{E}_k\|\widetilde{r}_{k+1}^\zeta\|_2^2. \tag{91}$$

Their refresh–transport covariance is

$$C_{k+1} := \mathbb{E}_k\langle \widetilde{r}_{k+1}^\zeta, z_{k+1}^\zeta\rangle. \tag{92}$$

The three moments have distinct roles. The fresh-gradient variance V_{k+1} measures the size of the noise that enters the estimator at the new point; the transport-noise moment T_{k+1} measures how much the component gradient changes between x_k and x_{k+1}; and the covariance C_{k+1} measures their alignment. Positive C_{k+1} means that transport noise $\widetilde{r}_{k+1}^\zeta$ and fresh-gradient noise z_{k+1}^ζ reinforce one another when they share the sample ζ. Negative C_{k+1} means that they partly cancel. Their magnitudes necessarily satisfy $|C_{k+1}| \le \sqrt{T_{k+1}V_{k+1}}$.

With iteration subscripts restored, $D_{k+1}(\tau)$ is the exact MARS-minus-VENUS innovation-variance difference in Eq. (13); positive values favor VENUS. For $V_{k+1} > 0$, its maximizer τ_{var}^* is given by Eq. (75). Appendix E provides the derivation. In particular, MARS has lower innovation variance than VENUS configured with the half split $\tau = 1/2$ if and only if

$$\frac{C_{k+1}}{V_{k+1}} < -\frac{\beta}{2(1-\beta)}. \tag{93}$$

Equation (13) separates the two mechanisms. The independent-refresh term $2\beta^2\tau(1-\tau)V_{k+1}$ always lowers VENUS's innovation variance and is largest at the half split. The covariance term $2\beta(1-\beta)\tau C_{k+1}$ helps VENUS when $C_{k+1} > 0$ and helps MARS only below the threshold in Eq. (93). This is a variance comparison; its connection to objective progress is tested in Appendix G.

Equation (13) contains both covariance and refresh averaging: at $\tau = 1/2$ it averages two independent batch-b refresh gradients. The compute-matched interpretation and direct BatchNorm-aware controls appear below and in Appendix G.

We compute V_{k+1}, T_{k+1}, and C_{k+1} exactly over all M_b components of each fixed minibatch oracle. Each oracle seed defines its own objective $f_{o,b}$, and the reported moments are exact conditional moments at the states visited during training.

F.3 Negative covariance and the half-split threshold

Across 544 matched-update trajectories and 5,440 measured state pairs, both signs of the covariance C_{k+1} occur. Negative covariance is uncommon at the state level: $156/5{,}440 = 2.87\%$ of measured states have $C_{k+1} < 0$. Because these events are distributed across trajectories, however, $130/544 = 23.9\%$ of trajectories contain at least one negative state. The state-level rate ranges from 1.79% to 3.63% across the six oracle–method groups. Negative covariance creates beneficial same-sample

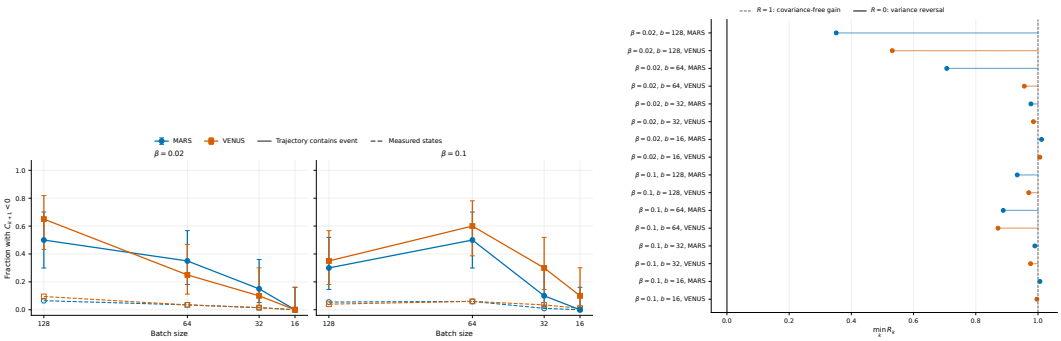
cancellation, but no state crosses the reversal threshold in Eq. (93): every state has $D_{k+1}(1/2) > 0$. To quantify the distance from reversal, define the retained-benefit ratio

$$R_{k+1} := \frac{D_{k+1}(1/2)}{\beta^2 V_{k+1}/2} = 1 + \frac{2(1-\beta)}{\beta}\frac{C_{k+1}}{V_{k+1}}. \tag{94}$$

It is also useful to subtract the covariance-free reference and write

$$\rho_{k+1} := R_{k+1} - 1 = \frac{2(1-\beta)C_{k+1}}{\beta V_{k+1}}. \tag{95}$$

At the half split, ρ_{k+1} is exactly the covariance contribution divided by the independent-refresh contribution. Here $\rho = -2$ gives $\tau^*_{\text{var}} = 0$, $\rho = -1$ is the half-split variance-reversal boundary, $\rho = 0$ means that covariance has no effect, and $\rho = 2$ gives $\tau^*_{\text{var}} = 1$. Across all 5,440 measured states, the 5th, 50th, and 95th percentiles of ρ_{k+1} are 0.007, 0.044, and 0.371, respectively. At the median state, the covariance term is only 4.4% as large as the independent-refresh term. The upper tail is not negligible, and the rare negative states remain important for testing the reversal threshold, but the typical same-b advantage at $\tau = 1/2$ is driven primarily by independent-refresh averaging. Here $R_{k+1} = 1$ is the covariance-free reference and $R_{k+1} = 0$ is the reversal boundary; positive values favor VENUS in innovation variance. The most adverse measured state has $R_{k+1} = 0.351$: negative covariance removes 64.9% of the independent-refresh benefit there, but does not reverse its sign. Conversely, some positive-covariance states have $\tau^*_{\text{var}} = 1$, so full decoupling is locally variance optimal there. Figure 26 summarizes both how often trajectories enter a negative-covariance state and how close their most adverse state comes to reversing the half split.



(a) Negative-state frequencies.

(b) Retained half-split variance advantage.

Figure 26: **Negative covariance and the half-split threshold.** Left: negative refresh–transport covariance $C_{k+1} < 0$ as a function of batch size. Solid curves give the fraction of trajectories containing at least one negative measured state; dashed curves give the fraction of all measured states that are negative. Both panels use the reference oracle (seed 1234). Solid-curve error bars are Wilson 95% intervals over training seeds; the dependent checkpoint-level fractions are shown without error bars. Across all three oracles, 23.9% of trajectories contain a negative state, while only 2.87% of measured states are negative. Right: the minimum retained-benefit ratio R_{k+1} from Eq. (94) within each cell. All minima remain above the variance-reversal boundary.

At $\beta = .02$, negative-covariance trajectories become more frequent from $b = 16$ to $b = 128$ in all six oracle–method series (Table 18); four of six paired intervals exclude zero. The trajectory-median C_{k+1}/V_{k+1} changes much less, and at $\beta = .1$ the frequency ordering is not consistent across all series. Because ideal minibatch scaling cannot change the sign at a fixed state, this is a trajectory effect. The half-split ordering remains unchanged, leaving the LLM batch-size crossover unexplained by this mechanism alone.

Oracle	Method	$\Delta \Pr(\exists k : C_{k+1} < 0)$	$10^4 \Delta \operatorname{median}_k(C_{k+1}/V_{k+1})$
1234	MARS	.43 [.14, .86]	1.47 [.44,2.56]
1234	VENUS	.86 [.57,1.00]	.64 [−.17,1.42]
1235	MARS	.29 [.00, .57]	.60 [−.39,1.67]
1235	VENUS	.29 [.00, .57]	.91 [−.44,2.20]
1236	MARS	.57 [.14, .86]	−.13 [−1.80,1.46]
1236	VENUS	.57 [.29, .86]	.43 [−.73,1.96]

Table 18: **Low-β batch endpoint contrasts.** Differences are $b = 128$ minus $b = 16$. The first quantity is the fraction of trajectories that visit any state with $C_{k+1} < 0$; the second is the mean across seeds of each trajectory's median C_{k+1}/V_{k+1}. Brackets are 95% paired-bootstrap intervals over the seven common training seeds, with each oracle analyzed separately.

F.4 Compute-matched variance predictions

At $\tau = 1/2$, VENUS averages two independent batch-b refresh gradients. MARS at batch $1.5b$ is the compute-matched baseline, while MARS at $2b$ approximately matches the distinct data used by the refresh term. With

$$M_{k+1} := (1-\beta)^2 T_{k+1} + \beta^2 V_{k+1} + 2\beta(1-\beta) C_{k+1}, \tag{96}$$

ideal i.i.d. scaling predicts $D_{k+1}^{\text{work}} = D_{k+1} - M_{k+1}/3$ and $D_{k+1}^{\text{data}} = D_{k+1} - M_{k+1}/2$. At the half split these are positive in 42.1% and 0% of the $5{,}440$ states, respectively; the work-matched fraction is 0% at $\beta = .02$ and 84.1% at $\beta = .1$. Direct BatchNorm-aware measurements in Appendix G agree in sign with the work-matched prediction in $111/112$ states. The same-b advantage therefore reflects mostly refresh averaging and does not persist uniformly after matching compute.

The bank uses $\gamma = 1$. Holding its states fixed while replacing γ by g scales C_{k+1} linearly and T_{k+1} quadratically. The covariance ratio becomes $g\rho_{k+1}$ (median 0.006 at $g = .13$), and the predicted work-matched advantage is positive in at least 98.6% of states for $g \le .13$ (the range tuned in the main experiments), versus 42.1% at $g = 1$. This counterfactual mainly reflects how many samples feed the β-weighted refresh when g is small; it neither captures drift reduction nor predicts the states that another training run would visit. Appendix G tests the objective consequences under Euclidean and layerwise-LMO updates.

F.5 Changing the training split changes the visited states

To test trajectory feedback, we train with $\tau_{\text{train}} \in \{0,.25,.5,.75,1\}$ in a control regime $(b,\beta) = (64,.02)$ and a sensitive regime $(16,.1)$, using three oracles and seven paired seeds per arm (210 fresh trajectories). Before this intervention, the covariance map selected the sensitive cell as the one with the largest minimum absolute shift among cells whose shift kept the same sign under all oracles. From the remaining cells, it selected the control with the smallest maximum absolute shift. No selection trajectory is reused.

Unlike a frozen-state comparison, τ_{train} acts throughout training and may change the later states (x_k,m_k). In the sensitive regime, every nonzero split lowers the trajectory-average C_{k+1}/V_{k+1} relative to coupled training, most strongly near $\tau_{\text{train}} = .5$ (Figure 27). The control response is smaller and does not replicate across all three oracles. Decoupling changes the covariance environment subsequently visited in the sensitive regime; the effect is not uniform across regimes.

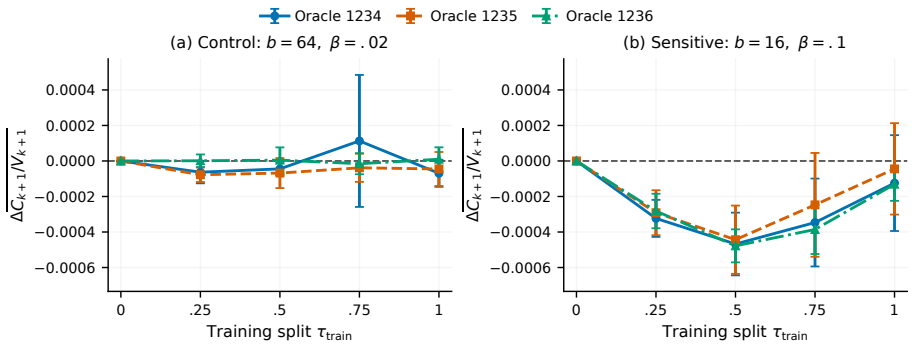


Figure 27: **Effect of the training split on normalized covariance C_{k+1}/V_{k+1}.** Points are paired changes relative to $\tau_{\text{train}} = 0$; error bars are 95% paired t intervals over seven seeds.

Table 19 decomposes the strongest case, $\tau_{\text{train}} = .5$. In the sensitive regime, V_{k+1} rises reproducibly while the raw change ΔC_{k+1} is unresolved. The normalized change is negative under every oracle, but it moves the half-split ratio by only $\Delta R_{k+1} = -0.0080$ to -0.0086. Training decoupling changes the noise scale of the visited states as well as their covariance; the normalized feedback is small relative to the reversal boundary $R = 0$.

Regime	ΔV_{k+1}	$10^4 \Delta(C/V)$	$10^3 \Delta R_{k+1}$
Control (64,.02)	0.21 to 0.30	−0.68 to 0.05	−6.7 to 0.5
Sensitive (16,.1)	0.97 to 1.03	−4.8 to −4.4	−8.6 to −8.0

Table 19: **Training-split moment decomposition at $\tau_{\text{train}} = .5$.** Ranges are the seven-seed paired means across the three oracle constructions, relative to $\tau_{\text{train}} = 0$. The raw ΔC_{k+1} is unresolved under every oracle in both regimes.

The exact baseline-state variance advantage $D_{k+1}(\tau)$ peaks at $\tau = .5$ in both regimes and under all three oracles. In training, $\tau_{\text{train}} = .5$ lowers the 99%-checkpoint finite-oracle loss relative to coupling by about .01 in the control regime and .02 in the sensitive regime; all six unadjusted paired 95% intervals exclude zero (Figure 28). In the control regime, the intervals overlap those of the other nonzero splits. The agreement with the variance calculation is only suggestive, because changing τ_{train} also changes the trajectory; moreover, $\tau = 1$ removes the independent-refresh term.

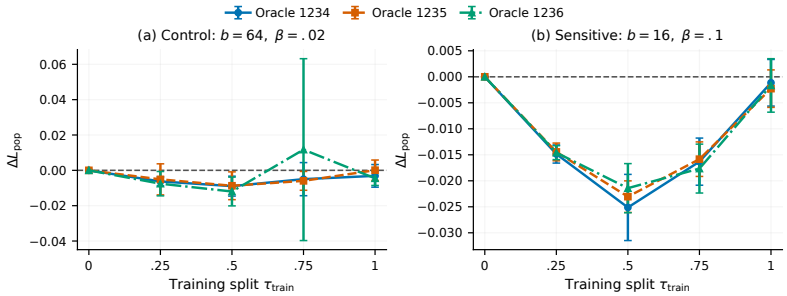


Figure 28: **Finite-oracle loss response to the training split.** The plotted quantity is $\Delta L_{\text{pop}} = L_{\text{pop}}(\tau_{\text{train}}) - L_{\text{pop}}(0)$; negative values favor the nonzero training split. Points are paired means and error bars are 95% paired t intervals over seven seeds, computed separately for each oracle. Loss is evaluated at 99% of training, the last predeclared diagnostic checkpoint rather than the final update. The point estimate is lowest at $\tau_{\text{train}} = .5$ in each regime–oracle comparison. This panel does not display $D_{k+1}(\tau)$; the exact variance response is computed separately from the coupled-baseline moments through Eq. (13).

In short, C_{k+1} varies across states. Its negative values never make $D_{k+1}(1/2) < 0$ here, while positive values can favor stronger decoupling. Appendix G fixes the state and varies only the probe split to test whether the resulting variance differences predict objective progress.

G FROZEN-STATE TEST OF THE VARIANCE MECHANISM

Summary. Appendix F shows that C_{k+1} changes along training, but changing τ also changes the trajectory. Here states are reached by training with MARS ($\tau_{\text{train}} = 0$) or VENUS ($\tau_{\text{train}} = .5$), then frozen while the probe value τ varies. Where VENUS's variance advantage $D_{k+1}(\tau)$ is larger, its Euclidean one-step objective advantage $A_1(k,\tau)$ is also larger. Under the layerwise LMO, VENUS at $\tau = .5$ beats the equal-work MARS@$1.5b$ baseline, but does not beat the higher-work MARS@$2b$ control (the interval crosses zero). The longer unequal-update comparison typically favors MARS and is treated as a secondary diagnostic. Throughout this appendix, positive differences favor VENUS.

G.1 PREDICTION AND OUTCOMES

We use the conditional moments V_{k+1}, T_{k+1} and C_{k+1} from Appendix F. With iteration subscripts restored, the centered innovations are those defined in Section 6, and their variance difference $D_{k+1}(\tau)$ is given by Eq. (13). At each fixed model state and incoming estimator, varying τ reallocates fresh-gradient weight from the shared sample ζ to the independent sample ξ, while leaving the transport contribution unchanged.

We measure two optimization outcomes. The *one-step finite-oracle advantage* is

$$A_1(k,\tau) := \mathbb{E}_k\big[f(x_{k+2}^{\text{MARS}}) - f(x_{k+2}^{\text{VENUS}}(\tau))\big], \tag{97}$$

where $f = f_{o,b}$ is the batch-indexed finite-oracle objective from Eq. (89). This comparison isolates the estimator effect after one update, using three gradient evaluations for VENUS and two for MARS. The *matched-work objective advantage* is

$$A_{\text{work}}(k,\tau) := \mathbb{E}\big[f(x_{\text{final}}^{\text{MARS}}) - f(x_{\text{final}}^{\text{VENUS}}(\tau))\big], \tag{98}$$

where MARS takes three updates and VENUS takes two, so both branches consume $6B$ per-example gradient evaluations at branch batch size B.

Both outcomes use the same finite-oracle objective, not the probe minibatches. The first isolates the local estimator effect; the second gives MARS an extra update so that the two short branches use the same gradient work.

The probe applies the estimator through the Euclidean update

$$x_{k+2}^{\text{branch}} = x_{k+1} - \eta m_{k+1}^{\text{branch}}. \tag{99}$$

We use the training value $\eta = .02$. This Euclidean probe is simpler than the layerwise LMO used in training, which we test separately below. Equal estimator means make its linear terms agree, but the loss difference depends on the curvature-weighted covariance $\text{tr}(\nabla^2 f_{o,b}(x_{k+1})\Sigma)$ rather than the unweighted $D_{k+1}(\tau)$. We use $D_{k+1}(\tau)$ as a predictor of the loss response.

To check whether the conclusion transfers to the optimizer used in the CNN experiments, we also apply the same layerwise LMO to each frozen estimator. The equal-work one-step outcome is

$$A_{1,\text{LMO}}^{\text{work}}(k,\tau) := \mathbb{E}_k\Big[f\Big(x_{k+2}^{\text{MARS@1.5}b,\text{LMO}}\Big) - f\Big(x_{k+2}^{\text{VENUS@}b,\text{LMO}}(\tau)\Big)\Big]. \tag{100}$$

Both branches then use $3b$ per-example gradient evaluations. We define $A_{1,\text{LMO}}^{\text{data}}$ by replacing MARS at $1.5b$ with MARS at $2b$. This second control approximately matches the number of distinct examples used by the two estimators and spends $4b$ gradient evaluations.

G.2 FROZEN-STATE INTERVENTION

The bank contains paired MARS and VENUS trajectories over seven batch sizes and four values of β. One paired seed configuration per cell gives 28 pairs and 56 trajectories, equally split by training origin. From each trajectory we select the two candidate checkpoints with the smallest and largest

paired R_{k+1}, without using an objective outcome. This gives 112 frozen states. At each state we hold the model and incoming estimator fixed and evaluate $\tau \in \{.25,.5,.75,1\}$ with 64 paired draws.

For the unequal-update matched-work diagnostic, an outcome-blind subset of 48 states balances the first, middle, and final thirds of training, quartiles of $D_{k+1}(1/2)$, and trajectory origin, with at most one state per trajectory. Within every replicate, MARS and all VENUS arms reuse the same ζ and ξ draws, so only the estimator construction changes.

Training follows Appendix F (Table 17), extended to $b \in \{16,32,64,128,256,512,1000\}$ and $\beta \in \{.02,.05,.1,.2\}$. Table 20 gives the remaining probe details.

Item	Setting
Checkpoint bank	7 batch sizes, 4 values of β, 3 oracles, 7 seeds
Training arms	$\tau_{\text{train}} \in \{0,.5\}$; 11,250 updates; 3 calls/update
Selected units	28 paired seed–configuration units; 112 frozen states
Probe arms	$\tau \in \{.25,.5,.75,1\}$; 64 paired draws/state
Sampler	Independent oracle indices with replacement; common random numbers
Evaluation	Global Euclidean moments and full finite-oracle loss $f_{o,b}$

Table 20: Frozen-state intervention protocol. The 112 states are selected without using A_1 or A_{work}.

The 28 paired seed–configuration units are the independent units. All intervals resample a complete pair, keeping both origins, both checkpoints, and all four values of τ together. Marginal rates describe this selected bank; independently selected random checkpoints check the equal-work LMO result.

Replay reproduces V_{k+1} and finite-oracle loss closely (median relative differences 0.09% and below 0.01%), but single-step T_{k+1} and C_{k+1} are more sensitive: seven of 112 states change the sign of C_{k+1}. We did not isolate the source. Every branch nevertheless uses the same replayed state and moments, and the main within-state slope reproduces (2.84×10^{-3} versus 2.72×10^{-3} originally).

G.3 Compute-matched larger-batch controls

Table 21 separates estimator MSE from objective progress and Euclidean from layerwise LMO geometry. Estimator MSE is measured relative to the common batch-b oracle gradient, so it includes the small BatchNorm-induced change in the estimator mean. All rows use the same 84 predeclared states from 21 paired configurations with complete LMO data, and intervals resample the complete pair.

On the full 112-state validation bank, measured MSEs for MARS@$1.5b$ and MARS@$2b$ match the ideal-scaling predictions: their median ratios are 1.000 and 0.998, all ratios lie in $[0.9,1.1]$, and the BatchNorm mean shift is below 5% of the error at every state. Measured and predicted signs of the compute-matched gap agree in $111/112$ states.

Comparison	States	Mean difference [95% CI]	Positive
Estimator MSE vs. MARS@$1.5b$	84	-2.98×10^{-3} $[-6.31,.410]\times10^{-3}$	46.4%
Estimator MSE vs. MARS@$2b$	84	-1.16×10^{-2} $[-1.79,-.663]\times10^{-2}$	0%
Euclidean objective vs. MARS@$1.5b$	84	-3.12×10^{-5} $[-5.78,-.903]\times10^{-5}$	59.5%
Layerwise LMO objective vs. MARS@$1.5b$	84	$+7.79\times10^{-5}$ $[5.05,10.5]\times10^{-5}$	85.7%
Layerwise LMO objective vs. MARS@$2b$	84	-1.32×10^{-5} $[-3.12,.539]\times10^{-5}$	20.2%

Table 21: Frozen-state controls at $\tau = .5$. Positive values favor VENUS. MARS at $1.5b$ matches the $3b$ gradient work of VENUS at b; MARS at $2b$ is the higher-work, approximately distinct-data-matched control. Coverage comprises $5/7$ batch cells at $\beta = .02$, $6/7$ at .05, and $5/7$ each at .1 and .2.

The equal-work conclusion depends on the update geometry. Estimator MSE is unresolved, while the Euclidean objective slightly favors MARS@$1.5b$ because of a few large differences at late, large-batch states. Under the layerwise LMO used in training, VENUS at $\tau = .5$ instead wins in 85.7% of the

same 84 states; independently selected random checkpoints give mean advantage 6.07×10^{-5} with 95% interval $[2.29,9.70] \times 10^{-5}$. The τ profile uses all 132 completed LMO states from 23 paired configurations. It follows the BatchNorm-aware compute-matched MSE gap measured on the 112-state validation bank; under ideal scaling, that gap is predicted by $D^{\text{work}}_{k+1}(\tau) = D_{k+1}(\tau) - M_{k+1}/3$. Both profiles peak at $\tau = .5$ and reach their minimum at $\tau = 1$. Yet the mean variance gap is negative at every τ, whereas the LMO advantage is positive for $\tau \in \{.25,.5,.75\}$. Estimator variance alone does not explain the LMO benefit. First-order alignment favors VENUS in 63.1% of states, but its mean interval crosses zero. The higher-work MARS@$2b$ comparison is unresolved. These controls support spending the third gradient on an independent refresh under the training geometry in these states. They do not establish a universal advantage for decoupling.

G.4 Within-state relation between $D_{k+1}(\tau)$ and $A_1(k,\tau)$

Unless stated otherwise, this subsection uses the 112 replayed states of Table 20. The phase analysis in Figure 30 and the best-τ comparison use the original checkpoint bank.

The primary analysis compares different τ values at the *same* frozen state:

$$A_1(k,\tau) = \alpha_k + \lambda D_{k+1}(\tau) + \varepsilon_{k,\tau}, \tag{101}$$

where the state effect α_k gives each frozen state its own intercept. The estimated slope is positive,

$$\lambda = 2.84 \times 10^{-3}, \qquad 95\%\ \text{CI} = [1.13,4.48] \times 10^{-3}.$$

At the same checkpoint, probe values with larger $D_{k+1}(\tau)$ tend to have larger $A_1(k,\tau)$. We obtain this estimate by subtracting each state's four-τ mean from both variables; intervals resample the paired training configuration.

Identification checks. Because $D_{k+1}(\tau)$ is a deterministic function of τ, the slope could merely reflect a generic preference for intermediate splits. Adding probe-τ fixed effects leaves it unchanged: 2.84×10^{-3} with 95% interval $[0.991,4.53] \times 10^{-3}$.

The pair $\tau = .25$ and $.75$ gives a more direct check of the covariance term. Their independent-refresh contributions are equal, so

$$D_{k+1}(.75) - D_{k+1}(.25) = \beta(1-\beta)C_{k+1}. \tag{102}$$

The observed mean $A_1(.75) - A_1(.25)$ is 6.39×10^{-6}, with pair-bootstrap interval $[1.00,13.0] \times 10^{-6}$, close to the predicted $5.94\,[1.36,14.0] \times 10^{-6}$. Across states it also increases with $\beta(1-\beta)C_{k+1}$ (slope $4.37\,[1.42,5.27] \times 10^{-3}$). These checks make a generic τ effect unlikely, although they cannot exclude all higher-moment effects.

The pooled Spearman correlation is 0.641, but it mixes states with different noise scales and curvature. Equation (101) is the more direct test because it uses only variation from changing τ at the same state; local curvature can still change the magnitude of the loss response.

Figure 29 likewise centers both variables within state. From the lowest to highest bin, mean centered A_1 rises from -2.76×10^{-5} to 2.43×10^{-5} and the positive fraction from 16.7% to 90.0%. The rise persists across the interior-τ bins, so it is not solely a $\tau = 1$ endpoint effect.

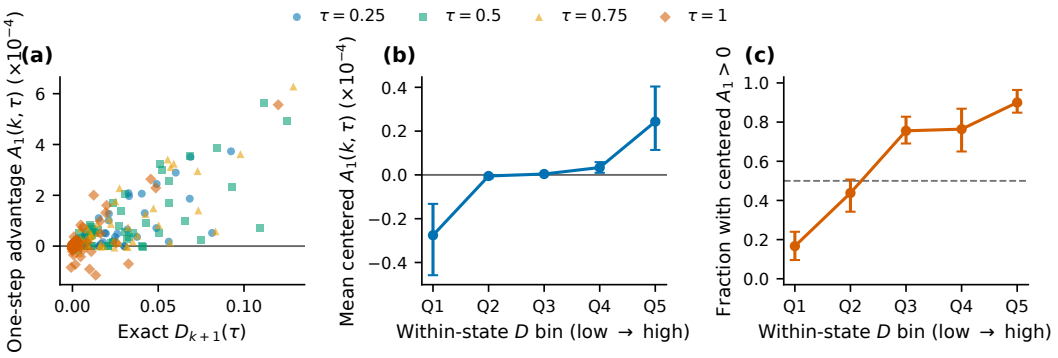
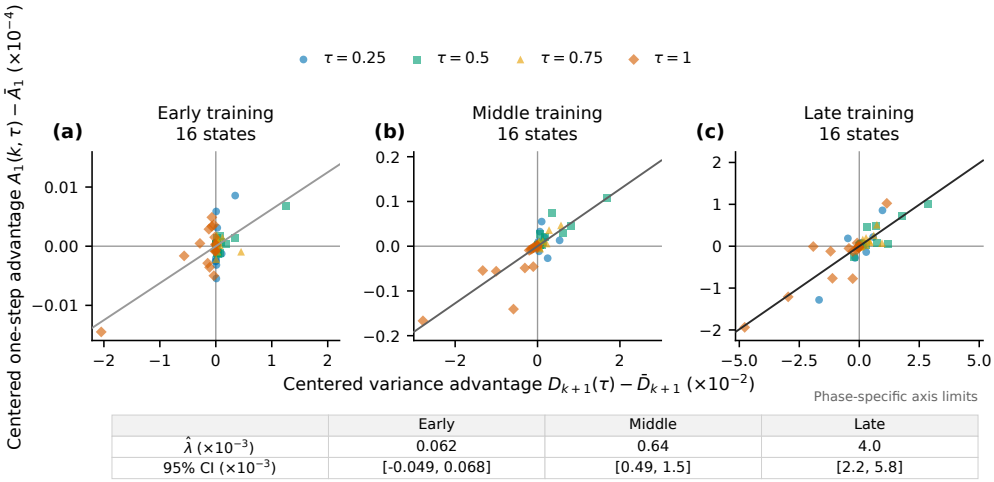


Figure 29: **Exact variance advantage $D_{k+1}(\tau)$ and the one-step objective advantage $A_1(k,\tau)$.** Left: all state–τ pairs. For the middle and right panels, both variables are centered by their four-τ mean within each state and divided into five equal-count bins of centered $D_{k+1}(\tau)$. Middle: mean centered A_1. Right: fraction with centered $A_1 > 0$. Error bars resample the complete paired training-seed configuration.

The relationship differs across training phases. We repeat Eq. (101) on 16 outcome-blind checkpoints from each third of training (Figure 30). The estimated slope is 0.062×10^{-3} in the first third, with an interval crossing zero. It is 0.64×10^{-3} in the middle third and 4.0×10^{-3} in the final third. This subset shows little resolved association early in training, a positive association in the middle third, and a substantially larger one in the final third. The early, middle, and late groups use different training trajectories, so the comparison does not follow the same run from one phase to the next.



	Early	Middle	Late
$\hat{\lambda}$ (×10⁻³)	0.062	0.64	4.0
95% CI (×10⁻³)	[-0.049, 0.068]	[0.49, 1.5]	[2.2, 5.8]

Figure 30: **Within-state variance–objective relation by training phase.** Each panel contains 16 outcome-blind states and compares four values of τ after centering $D_{k+1}(\tau)$ and $A_1(k,\tau)$ within each state. Phase-specific axis limits keep all three point clouds visible, so the line angles are not directly comparable. The table reports the slopes and 95% intervals on a common scale. These are percentile pair-bootstrap intervals and need not be symmetric around the point estimate.

The D-maximizing and empirical A_1-maximizing grid values agree in 50.0% of states (95% interval [38.4,61.6]%, versus 25% by chance). However, the median simultaneous confidence set contains all four arms; among the 54 states where it excludes at least one arm, it retains the D-maximizer in 53. This supports the directional relation, while the objective-optimal τ usually remains unresolved at a single checkpoint.

G.5 Secondary matched-work comparison with unequal update counts

Equalizing work by comparing three MARS updates with two VENUS updates does not favor VENUS in the typical state. At $\tau = .5$, the mean advantage is -2.92×10^{-5} with interval $[-2.33 \times 10^{-4}, 1.93 \times 10^{-4}]$, and $13/48 = 27.1\%$ of states favor VENUS (interval $[14.6, 39.6]\%$). This branch also changes the update count and path length, so the equal-update controls above are the primary evidence.

Figure 31 centers D_{k+1} and A_{work} within each state. From its lowest to highest centered-D bin, the mean centered outcome rises from -3.17×10^{-5} to 2.96×10^{-5} and the positive fraction from 12.8% to 87.2%. Larger within-state $D_{k+1}(\tau)$ marks more favorable short-branch outcomes; the branch combines the estimator change with a different update count.

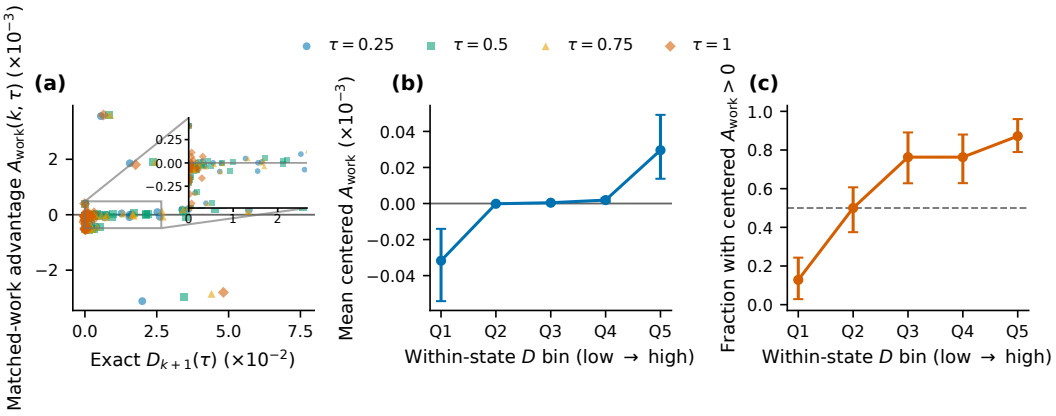


Figure 31: **Matched-work objective advantage $A_{\text{work}}(k,\tau)$.** MARS takes three updates and VENUS two. Left: A_{work} versus the exact variance advantage $D_{k+1}(\tau)$. For the middle and right panels, both variables are centered by their four-τ mean within each state and divided into five equal-count bins of centered $D_{k+1}(\tau)$. Middle: mean centered A_{work}. Right: fraction with centered $A_{\text{work}} > 0$. Error bars are 95% intervals from resampling the complete paired training configuration.

Relation to the batch-size pattern. With only one to four paired configurations per batch, this bank cannot estimate a batch trend. Appendix F explains why ideal fixed-state scaling changes the magnitude, but not the sign, and why the $\gamma = 1$ bank differs from the small-γ GPT-2 runs. The main complexity bounds permit a small-batch advantage without predicting monotonicity or a crossover. Their favorable $(1 - \gamma)\varepsilon^{-4}$ term vanishes here, so it cannot explain the local LMO result.

Summary. At a fixed state, larger $D_{k+1}(\tau)$ predicts larger Euclidean one-step $A_1(k,\tau)$. Under the layerwise LMO, VENUS at $\tau = .5$ also improves over equal-work MARS@$1.5b$, but not over the higher-work MARS@$2b$ control. The unequal-update matched-work branch does not favor VENUS on average; larger $D_{k+1}(\tau)$ only identifies states with more favorable short-branch outcomes.

H Gradient-noise covariance in Transformer training

Appendix E shows that refresh–transport covariance can change the variance-optimal amount of decoupling at a fixed state. Appendix F measures these moments for finite-population CIFAR-10 oracles, and Appendix G studies their local optimization consequences. Here we examine the corresponding gradient-noise estimates along Transformer training trajectories.

H.1 Conditional variance decomposition

We use the conditional fresh-gradient variance V_{k+1} and refresh–transport covariance C_{k+1} defined in Appendix E, retaining the iteration subscripts omitted in Section 6. Their MARS-minus-VENUS innovation-variance difference $D_{k+1}(\tau)$ is given by Eq. (13): the independent-refresh term is non-negative, whereas the covariance contribution can have either sign. For $V_{k+1} > 0$, Eq. (75) gives

the variance-minimizing split τ^*_{var}. Below we estimate these quantities per coordinate within each Transformer block.

H.2 Training protocol and estimators

At branch batch size 16 and an equal budget of 2.699B gradient tokens, the selected exact VENUS configuration achieves mean validation loss 3.5849, compared with 3.5958 for exact MARS, over confirmation seeds 53–58. The paired difference is -0.0109 with a 95% confidence interval $[-0.0150, -0.0068]$, and VENUS wins on all six seeds (Table 10). This performance comparison does not by itself identify the responsible variance mechanism.

We instrument seeds 53–55, the first three consecutive seeds from the six-seed exact-VENUS confirmation experiment reported in Table 10. These seeds were fixed independently of their diagnostic outcomes. The model is a 124M-parameter GPT-2-scale Transformer trained on FineWeb with context length 1024, branch batch size 16, 54,912 optimizer steps, and effective parameters $(\beta, \gamma, \tau) = (0.02, 0.0015, 0.5)$. Each run consumes 2.699B gradient tokens and 1.799B fresh-token exposures, counting each sampled minibatch once.

At a sampled step, let $g^0_{\ell,k+1}$ and $g^+_{\ell,k+1}$ denote the gradients of Transformer block ℓ at x_k and x_{k+1} on the same transport minibatch ζ_{k+1}. Let $h^+_{\ell,k+1}$ be the current-point gradient on the independent refresh minibatch ξ_{k+1}. If d_ℓ is the number of coordinates in block ℓ, we record

$$\widehat{V}_{\ell,k+1} := \frac{\|g^+_{\ell,k+1} - h^+_{\ell,k+1}\|_2^2}{2d_\ell}, \tag{103}$$

$$\widehat{C}_{\ell,k+1} := \frac{\gamma}{d_\ell}\left\langle g^+_{\ell,k+1} - g^0_{\ell,k+1}, g^+_{\ell,k+1} - h^+_{\ell,k+1}\right\rangle, \tag{104}$$

$$\widehat{D}_{\ell,k+1} := 2\beta\tau\left[\beta(1-\tau)\widehat{V}_{\ell,k+1} + (1-\beta)\widehat{C}_{\ell,k+1}\right]. \tag{105}$$

Conditional on a fixed pair of states and equal-law independent minibatches, the first two quantities are unbiased for the per-coordinate versions of V_{k+1} and C_{k+1}. The third is their plug-in MARS-minus-VENUS innovation-variance difference. The diagnostic reuses the three gradients already required by exact VENUS and adds no forward or backward evaluation. We sample every ten steps, obtaining 5,491 observations per block and seed, or 16,473 observations per block in total.

H.3 Phase and block results

Table 22 reports means and standard errors over the three seeds. The mean covariance proxy changes sign from positive in the first third of training to negative in the middle and final thirds. This sign pattern holds separately for all three seeds. Its magnitude is nevertheless too small to reverse the total variance comparison: $\widehat{D}_{\ell,k+1} > 0$ for all 197,676 sampled block–step pairs. In the early phase the covariance contribution accounts for approximately 1.6% of $\widehat{D}$; in the middle and final phases it is negative but below 0.06% in magnitude. Consequently, the mean plug-in variance-optimal split moves from 0.50409 early to values essentially equal to 0.5 later.

Phase	$\widehat{V}$	$\widehat{C}$	$\widehat{D}$	$\widehat{\tau}^*_{\mathrm{var}}$
Early	$(2.023\pm0.098)\times10^{-8}$	$(3.413\pm0.539)\times10^{-12}$	$(4.112\pm0.207)\times10^{-12}$	0.50409 ± 0.00046
Middle	$(2.058\pm0.024)\times10^{-10}$	$(-1.225\pm0.039)\times10^{-15}$	$(4.114\pm0.049)\times10^{-14}$	0.499854 ± 0.000006
Late	$(2.584\pm0.026)\times10^{-10}$	$(-0.557\pm0.073)\times10^{-15}$	$(5.166\pm0.052)\times10^{-14}$	0.499947 ± 0.000007

Table 22: Per-coordinate plug-in estimates aggregated over all 12 Transformer blocks, reported as mean $\pm$ SEM over three seeds. The phases are equal thirds of training. Positive $\widehat{D}$ favors VENUS in the estimated innovation second moment.

Block	$\widehat{C}_{\mathrm{early}}/10^{-12}$	$\widehat{C}_{\mathrm{mid}}/10^{-15}$	$\widehat{C}_{\mathrm{late}}/10^{-15}$	$\widehat{D}_{\mathrm{all}}/10^{-12}$
0	$+4.624$	-6.311	-3.233	1.867
1	$+4.208$	-0.711	-0.326	1.497
2	$+4.101$	-0.856	-0.405	1.421
3	$+3.827$	-0.788	-0.368	1.363
4	$+3.535$	-0.626	-0.257	1.318
5	$+3.323$	-0.572	-0.222	1.304
6	$+3.179$	-0.470	-0.176	1.288
7	$+3.150$	-0.457	-0.170	1.301
8	$+2.806$	-0.403	-0.130	1.272
9	$+2.888$	-0.416	-0.138	1.279
10	$+2.712$	-0.605	-0.207	1.267
11	$+2.602$	-2.486	-1.057	1.374

Table 23: Transformer-block results averaged over three seeds. Covariance columns give phase means; $\widehat{D}_{\mathrm{all}}$ is the trajectory-averaged plug-in MARS-minus-VENUS innovation-variance difference. All quantities are per coordinate.

The phase-level sign pattern is coherent across depth and seeds: all 12 blocks in each of the three trajectories have positive early mean $\widehat{C}$ and negative middle and final means. The absolute value of $\widehat{D}_\ell$ is largest in the first blocks and smallest near block 10, but $\widehat{D}_\ell/\widehat{V}_\ell$ varies only from 2.0246×10^{-4} to 2.0378×10^{-4}. Thus no single Transformer block drives the estimated advantage.

The realized plug-in covariance is positive in only 14%–38% of sampled steps, depending on the block. This sample-level frequency must not be identified with the fraction of states having exact conditional $C_{k+1} > 0$: unlike Appendix F, we do not repeatedly evaluate the oracle at each frozen state. The positive full-run mean instead reflects the larger magnitude of the early positive observations.

H.4 Scope and limitations

This diagnostic is descriptive. It follows three trajectories composed of time-correlated states, and it measures Euclidean/Frobenius gradient-space quantities before the spectral LMO. Unlike the intervention in Appendix G, it does not vary τ while holding the state and incoming estimator fixed. It therefore does not establish that the measured variance difference causes better optimization. The supported conclusion is narrower: across three prespecified consecutive seeds, the phase-averaged refresh–transport covariance proxy changes sign coherently across Transformer blocks, while the independent-refresh term keeps the estimated total innovation-variance advantage positive at every sampled block–step pair.

I Conclusion

We introduced VENUS by decoupling the minibatches in the MSGD–MVR representation of the unclipped MARS estimator. The analysis and empirical studies support a balanced split as a practical default, without establishing its universal optimality. Under matched gradient budgets, exact VENUS achieves lower final validation loss at LLM batch size 16, the difference at 32 is inconclusive, and exact MARS performs better at 64. The diagnostics distinguish local variance reduction from gains after accounting for computational cost, while the stronger performance of the tested cached-gradient MARS variant limits broader claims of practical superiority. These results support sample independence as a regime-dependent design choice and leave prediction of its full-training benefit as an open question.