

Machine-Checked Scalar Foundations for SGD Analysis

A Reusable Corpus of Conditional Algebraic Lemmas

Tamás Nagy, Ph.D.

tnagyphd@gmail.com

Working Paper • July 2026

Build-std v2.0+6788810519 | 2026-07-20 11:21 | 738cb35c

Abstract

We present a machine-checked corpus of 78 real-arithmetic theorems that occur as local steps in standard analyses of stochastic gradient descent (SGD). The corpus covers sign and ordering facts, scalar forms of convexity and smoothness consequences, one-step distance rearrangements, elementary momentum and mini-batch algebra, assumed control-variate bounds, rate-shape inequalities, schedule monotonicity, and regularization terms. Forty-three selected results are exposed in the paper; the remaining declarations are support and recovery lemmas. The verification replay checks 109 registered items: 31 real-symbol registrations and 78 theorem declarations. The kernel introduces no domain axioms, but every theorem remains conditional on the assumptions displayed in its statement. In particular, the development does not define an SGD process, a probability space, a convex function, or an arbitrary-horizon iteration, and it does not prove end-to-end convergence of SGD or its variants. This is an artifact and corpus report, not a claim of new SGD mathematics; its contribution is transparent organization, replayability, and theorem-level traceability for a scalar foundation used in such analyses.

1. Introduction

1.1 The Problem

Stochastic gradient descent is a standard optimization method in modern machine learning. Its convergence theory is classical, but textbook arguments repeatedly depend on small algebraic transformations whose hypotheses can be obscured by surrounding functional and probabilistic notation. This paper isolates a scalar layer of those arguments and checks each implication mechanically.

1.2 Main Results

This paper presents a machine-checked scalar lemma corpus supporting standard SGD analyses. Its central scope claim is:

Corpus claim (informal). *Under the scalar premises displayed in each statement, the kernel verifies one-step, rearrangement, sign, monotonicity, and rate-shape consequences used in familiar analyses of SGD and related methods.*

The paper-facing selection covers:

1. **Descent step bounds:** scalar sign and ordering facts (§2)
2. **Convexity-shaped inequalities:** consequences of supplied scalar premises (§3–§4)

3. **Smoothness-shaped descent algebra**: consequences of supplied one-step bounds (§5)
4. **Distance and noise rearrangements**: one-step scalar implications (§6)
5. **Momentum and variance-reduction algebra**: elementary consequences, not algorithmic convergence (§7–§8)
6. **Rate and schedule shapes**: reciprocal and contraction inequalities (§9–§10)
7. **Averaging, scaling, and regularization terms**: scalar identities and inequalities (§11–§12)

1.3 Proof Architecture

The source groups theorems by their intended analytical role. These labels provide organization; they do not add function-space or probabilistic semantics absent from the formal statements:

1. **Real arithmetic layer** — Kernel-checked equalities and inequalities over
2. **Descent primitives** — Basic step bounds (`descent_step`, `loss_decreases`)
3. **Convexity layer** — Suboptimality from convexity (`suboptimality_from_convexity`)
4. **Strong convexity** — Quadratic growth (`strong_convex_quadratic_growth`)
5. **Stochastic layer** — Noise bounds (`second_moment_upper_bound`, `noise_term_nonneg`)
6. **Method-shaped algebra** — Momentum, variance-bound, and schedule consequences

1.4 Formalization Statistics

Metric	Value
Theorem declarations	78
Registered real symbols	31
Paper-facing selected results	43
Domain axioms introduced by the field	0
Total registered items checked	109
Errors	0
Proof files	1

The counting rule is disjoint: the 109 replayed items consist of 31 registered real symbols plus 78 theorem declarations. The 43 paper-facing results are a selected subset of the 78 theorems, not an additional category to add to the total.

1.5 Notation and Semantic Convention

Every symbol in a displayed theorem denotes a real scalar in the source. Labels such as $\|f\|^2$, f , $x - x$, $E[\|g\|^2]$, Var , and $\|x_t - x\|^2$ are mnemonic paper notation for those scalars; the source does not construct a normed space, gradient, expectation, variance, function, or iterate sequence. Subscripts label one-step quantities and are not quantified time indices. Consequently, the displayed formulas are paper-level substitutions into the associated scalar declarations, not literal transcriptions of richer formal objects.

Paper notation	Source scalar or expression	Status of the bridge
$\ f\ ^2$	<code>grad_sq</code>	mnemonic rename only
f , $x - x^*$	<code>grad_inner</code>	mnemonic rename only

Paper notation	Source scalar or expression	Status of the bridge
$(\cdot/2)\ x - x^*\ ^2$	<code>mu_half * dist_sq</code>	<code>mu_half</code> and <code>dist_sq</code> are independent real scalars; no norm or $\cdot/2$ definition is checked
$(L^2/2)\ f\ ^2$	<code>L_half_eta_sq * grad_sq</code>	independent scalar factor; no checked composite definition
$E[\ g\ ^2]$ and Var	<code>second_moment</code> and <code>variance</code>	independent real scalars related only by the displayed theorem premise
$\ x_t - x\ ^2, \ x_{\{t+1\}} - x\ ^2$	<code>dist_curr</code> , <code>dist_next</code>	one-step scalar labels, not members of a sequence

The exact checked statements are therefore the source expressions named in `paper_claim_map.json`. The optimization-shaped notation records an intended interpretation and does not add a formal bridge theorem.

2. Descent Step Bounds

This section records elementary sign and ordering facts used inside scalarized descent arguments. It does not formalize the analytic descent lemma for a smooth function.

Theorem 2.1 (`descent_step`). *For scalars $\epsilon > 0$, $\|f\|^2 = 0$, and $L > 0$, if $\epsilon \leq L$, then $\epsilon \cdot \|f\|^2 = 0$.*

This establishes only the non-negativity of the scalar product. The source premise `L` is unused for that conclusion and is not the conventional smoothness restriction `c/L`; this theorem supplies no learning-rate validity or progress guarantee.

Theorem 2.2 (`loss_decreases`). *For any decrease $d = 0$, if $f_{\{t+1\}} = f_t - d$, then $f_{\{t+1\}} \leq f_t$.*

Theorem 2.3 (`smaller_lr_smaller_step`). *For $\|f\|^2 = 0$ and $0 \leq \epsilon$, we have $\epsilon \cdot \|f\|^2 = 0$.*

This monotonicity compares two scalar products; validity of an optimization step is outside the theorem.

Theorem 2.4 (`zero_grad_stationary`). *If $\|f\|^2 = 0$, then $\epsilon \cdot \|f\|^2 = 0$.*

At stationary points, gradient steps have zero magnitude.

Theorem 2.5 (`lr_grad_product_nonneg`). *For $\epsilon = 0$ and $\|f\|^2 = 0$, we have $\epsilon \cdot \|f\|^2 = 0$.*

3. Convexity Analysis

The kernel does not define convex functions. It accepts scalar inequalities corresponding to familiar first-order convexity consequences and verifies their rearrangements.

Theorem 3.1 (`suboptimality_from_convexity`). *If $f_t \leq f$, $x - x^* \leq f$, then $f_t - f \leq f$, $x - x^* \leq f$.*

Given the supplied scalar premise, the conclusion is its algebraic rearrangement.

Theorem 3.2 (`nonneg_suboptimality`). *If $f \leq f_t$, then $0 \leq f_t - f$.*

Theorem 3.3 (scaled_suboptimality). *If $f_t - f, x - x^* \leq \epsilon$ and $\epsilon \geq 0$, then $(f_t - f) \leq \epsilon \cdot f, x - x^*$.*

Theorem 3.4 (double_scaled_suboptimality). *If $f_t - f, x - x^* \leq \epsilon$ and $\epsilon \geq 0$, then $2(f_t - f) \leq \epsilon \cdot f, x - x^*$.*

4. Strong Convexity

The kernel does not define strong convexity. This section verifies consequences of supplied scalar inequalities shaped like strong-convexity bounds.

Theorem 4.1 (strongly_convex_implies_convex). *For $\mu/2 > 0$, $\|x - x^*\|^2 \leq 0$, if $f_t + \mu, x^* - x + (\mu/2)\|x - x^*\|^2 \leq f$, then $f_t + \mu, x^* - x \leq f$.*

The non-negative quadratic term can be dropped from the supplied scalar bound.

Theorem 4.2 (strong_convex_quadratic_growth). *If $f + (\mu/2)\|x - x^*\|^2 \leq f_t$, then $(\mu/2)\|x - x^*\|^2 \leq f_t - f$.*

The conclusion is an algebraic rearrangement of the supplied quadratic-growth premise.

Theorem 4.3 (strong_convex_tighter_suboptimality). *If $f_t - f, x - x^* + (\mu/2)\|x - x^*\|^2 \leq f$, then $f_t - f^* \leq f, x - x^* - (\mu/2)\|x - x^*\|^2$.*

5. Gradient Lipschitz and Stochastic Bounds

The kernel does not define Lipschitz gradients. It verifies consequences after the relevant scalar one-step and comparison inequalities are supplied as premises.

Theorem 5.1 (gradient_step_descent). *If $f_{t+1} \leq f_t - \|f\|^2 + (L^2/2)\|f\|^2$ and $(L/2)\|f\|^2 \leq \|f\|^2$, then $f_{t+1} \leq f_t$.*

Theorem 5.2 (stochastic_step_upper_bound). *For scalars bearing stochastic-gradient labels, if $f_{t+1} \leq f_t - f, g + (L^2/2)\|g\|^2, f, g \leq 0$, and $(L^2/2)\|g\|^2 \leq f, g$, then $f_{t+1} \leq f_t$.*

Theorem 5.3 (second_moment_upper_bound). *If $\text{Var} \leq 0$ and $E[\|g\|^2] = \|f\|^2 + \text{Var}$, then $\|f\|^2 \leq E[\|g\|^2]$.*

Theorem 5.4 (noise_term_nonneg). *For $\epsilon \geq 0$ and $\epsilon^2 \geq 0$, we have $\epsilon^2 \geq 0$.*

6. Distance-Bound Rearrangements

The symbols in this section are real scalars labeled as squared distances. There is no formal sequence of iterates; the results are one-step rearrangements.

Theorem 6.1 (suboptimality_from_distance). *If $\|x_{t+1} - x\|^2 \leq \|x_t - x\|^2 - 2(f_t - f) + \epsilon^2$, then $2(f_t - f) \leq \|x_t - x\|^2 - \|x_{t+1} - x\|^2 + \epsilon^2$.*

Theorem 6.2 (drop_final_distance). *If $\|x_{t+1} - x\|^2 \leq 0$ and the distance bound holds, then $2(f_t - f) \leq \|x_t - x\|^2 + \epsilon^2$.*

Theorem 6.3 (contraction_preserves_bound). *Under strong convexity with $\mu \geq 0$, $\epsilon \geq 0$, if $\|x_{t+1} - x\|^2 \leq \|x_t - x\|^2 - \|x_t - x\|^2 + \epsilon^2$ and $\|x_t - x\|^2 \leq 0$, then $\|x_{t+1} - x\|^2 \leq \|x_t - x\|^2 + \epsilon^2$.*

Theorem 6.4 (gradient_dominates_gap). For $\beta > 0$, $f_t - f \leq 0$, if $2(f_t - f) \leq \|f\|^2$, then $(f_t - f) \leq \|f\|^2$.

Theorem 6.5 (condition_number_ge_one, legacy source identifier). For $\beta > 0$, $L > 0$, $\beta \leq L$, we have $\beta \leq L$.

This theorem directly repeats one premise. The identifier is retained only for exact source traceability; it neither defines a condition number nor proves a lower bound for one.

7. Momentum-Shaped Algebra

These results establish elementary sign and upper-bound facts for scalar expressions labeled as momentum terms. Here β is a scalar discount factor constrained to $[0,1]$. The results do not prove acceleration or heavy-ball convergence.

Theorem 7.1 (momentum_accumulation_nonneg). For $0 \leq \beta \leq 1$, $m_{\text{prev}} \geq 0$, $\|f\| \geq 0$, we have $\beta \cdot m_{\text{prev}} + \|f\| \geq 0$.

Theorem 7.2 (momentum_bounded_by_sum). For $0 \leq \beta \leq 1$, $m_{\text{prev}} \geq 0$, $\|f\| \geq 0$, we have $\beta \cdot m_{\text{prev}} + \|f\| \leq m_{\text{prev}} + \|f\|$.

Theorem 7.3 (heavy_ball_step_nonneg). For $\beta > 0$ and $m_t > 0$, we have $\beta \cdot m_t > 0$.

8. Mini-Batch and Assumed Variance Bounds

The first two results compare the scalar expression σ^2/B across positive batch sizes. The control-variate results assume the relevant variance ordering; they do not derive it from an estimator construction and do not justify a learning-rate change.

Theorem 8.1 (minibatch_variance_reduction). For $\sigma^2 > 0$ and batch size $B > 1$, we have $\sigma^2/B < \sigma^2$.

Theorem 8.2 (larger_batch_less_noise). For $\sigma^2 > 0$, $0 < B < B'$, we have $\sigma^2/B < \sigma^2/B'$.

Theorem 8.3 (batch_gradient_nonneg). For batch size $B > 0$ and batch gradient sum ≥ 0 , the average gradient is non-negative.

Theorem 8.4 (svrg_variance_bound). For $0 \leq \text{Var}\{SVRG\} \leq \text{Var}\{SGD\}$, we have $\text{Var}\{SGD\} - \text{Var}\{SVRG\} \geq 0$ and $\text{Var}\{SVRG\} \leq \text{Var}\{SGD\}$.

Theorem 8.5 (control_variate_reduces_variance). For $0 \leq \text{reduction} \leq \text{Var}\{\text{orig}\}$, we have $\text{Var}\{\text{orig}\} - \text{reduction} \geq 0$ and $\text{Var}\{\text{orig}\} - \text{reduction} \leq \text{Var}\{\text{orig}\}$.

9. Rate-Shape Algebra

This section verifies monotonicity of reciprocal and contraction-shaped scalar expressions. Here ρ is a scalar contraction parameter constrained to $[0,1]$. The section does not connect these expressions to an SGD iterate sequence.

Theorem 9.1 (sublinear_rate_decreases). For $C > 0$, $0 < T < T'$, we have $C/T < C/T'$.

The reciprocal expression C/T decreases with T ; no asymptotic bound for an algorithm is established.

Theorem 9.2 (linear_rate_contraction). For $0 < \alpha < 1$ and $gap > 0$, we have $\alpha \cdot gap < gap$.

Multiplication by a scalar in $[0,1)$ does not increase a non-negative quantity.

Theorem 9.3 (contraction_iterated_shrinks). For $0 < \alpha < 1$ and $gap > 0$, we have $\alpha^2 \cdot gap < \alpha \cdot gap$.

A second multiplication by the same scalar does not increase the expression. Arbitrary-horizon iteration is not formalized.

10. Learning Rate Schedules

These theorems concern scalar formulas commonly used as learning-rate schedules; they do not analyze a training process.

Theorem 10.1 (lr_decay_monotone). For $\alpha > 0$, $0 < t < t'$, we have $\alpha/t < \alpha/t'$.

Decay schedules produce monotonically decreasing learning rates.

Theorem 10.2 (warmup_lr_bounded). For $\alpha_{max} > 0$, $t > 0$, $T_{warmup} > 0$, $t \leq T_{warmup}$, we have $(\alpha_{max} \cdot t)/T_{warmup} \leq \alpha_{max}$.

Warmup schedules are bounded by the maximum learning rate.

Theorem 10.3 (warmup_lr_nonneg). For $\alpha_{max} > 0$, $t \geq 0$, $T_{warmup} > 0$, we have $(\alpha_{max} \cdot t)/T_{warmup} \geq 0$.

11. Polyak Averaging and Batch Size Scaling

The same scalar monotonicity patterns also appear in formulas labeled as averaging weights and batch-scaling rules.

Theorem 11.1 (polyak_average_convex_combination). For $T > 1$, the expression $((T-1)/T)x_{avg} + (1/T)x_{new}$ equals itself.

The current source theorem is a direct restatement. It does not prove that the weights are non-negative or sum to one, and it is retained to make that trust limitation explicit.

Theorem 11.2 (averaging_weight_decreases). For $1 < T < T'$, we have $1/T < 1/T'$.

The scalar weight $1/T$ decreases with T .

Theorem 11.3 (linear_scaling_rule). For $\alpha_{base} > 0$ and $k > 1$, we have $\alpha_{base} < k \cdot \alpha_{base}$.

This proves only the scalar ordering $\alpha_{base} < k \cdot \alpha_{base}$. It does not prove that a larger learning rate is permitted by an optimization method.

Theorem 11.4 (sqrt_scaling_intermediate). For $\alpha_{base} > 0$ and $1 < k_{sqrt} < k$, we have $\alpha_{base} < k_{sqrt} \cdot \alpha_{base} < k \cdot \alpha_{base}$.

The intermediate positive multiplier gives the stated scalar ordering; no batch-size rule is derived.

12. Gradient Noise and Proximal SGD

Finally, the corpus records scalar consequences of supplied bias–variance decompositions and regularization terms.

Theorem 12.1 (`biased_gradient_error_decomposition`). *If $MSE = bias^2 + variance$ with $bias^2 = 0$ and $variance = 0$, then $bias^2 = MSE$ and $variance = MSE$.*

Theorem 12.2 (`unbiased_estimator_mse_equals_variance`). *If $MSE = 0 + variance$, then $MSE = variance$.*

Theorem 12.3 (`proximal_step_nonneg`). *For $\alpha \geq 0$ and $\|x\|^2 \geq 0$, we have $\alpha \cdot \|x\|^2 \geq 0$.*

Theorem 12.4 (`regularization_shrinks_norm`). *For $\|x\| > 0$, $0 < \alpha < 1$, we have $(1 - \alpha)\|x\| < \|x\|$.*

These are scalar regularization-term facts. No proximal operator or proximal-SGD update is defined.

13. Discussion

13.1 Scope and Limitations

This formalization covers scalar real arithmetic only. It does not include:

- functions, gradients, norms, inner products, or convexity predicates;
- random variables, expectations, probability spaces, or variance definitions;
- an SGD update relation, iterate sequence, or arbitrary-horizon convergence theorem;
- derivations of momentum acceleration, SVRG variance reduction, Polyak averaging guarantees, batch-scaling validity, or proximal-SGD convergence;
- matrix/tensor adaptive methods, non-convex analysis, distributed SGD, or high-probability bounds.

13.2 Related Work

The scalar patterns are motivated by classical optimization analyses due to Nesterov, Bottou, Johnson and Zhang, Polyak, and Robbins and Monro [1–6]. The mathematical claims themselves are standard; this paper does not claim theorem-level originality.

Mechanized optimization already extends beyond the scope of this corpus. Li et al. formalize complexity results for gradient, subgradient, proximal-gradient, and accelerated methods in Lean 4 [7]. Vajjha et al. formalize Dvoretzky’s stochastic-approximation convergence theorem in Coq [8]. Isabelle/HOL has reusable foundations for unconstrained optimization and optimality conditions [9]. Most directly, Cassie’s 2026 sgd–lean artifact defines an SGDSetup over a real inner-product space, assumes a bounded-noise gradient model, proves a one-step distance inequality, and derives an averaged $O(1/\sqrt{K})$ bound in Lean 4 [10]. It therefore occupies the algorithmic and semantic layer that this corpus does not reach. This manuscript also supersedes an earlier working draft, *SGD Is Right: A Machine-Checked Proof That Stochastic Gradient Descent Converges*, built from the same 78-theorem artifact; the present scalar framing replaces, rather than independently corroborates, that draft’s overstrong convergence interpretation. The present contribution is neither a competing convergence result nor a claim of superiority: it is a narrower auditable inventory of small conditional scalar implications, explicit trust boundaries, and a claim map for possible reuse or later semantic lifting.

13.3 Future Directions

1. **Semantic lift:** replace scalar labels with functions, inner-product spaces, gradients, and convexity predicates.
2. **Stochastic lift:** define probability spaces, conditional expectations, unbiased estimators, and variance.
3. **Iterative lift:** define SGD trajectories and prove finite-horizon and asymptotic convergence results.
4. **Lean export hardening:** reduce the generated preamble neutralization surface and replace asserted declarations with narrower imported foundations where practical.

14. Conclusion

The verified artifact is a corpus of conditional real-arithmetic lemmas, not a formal convergence theory for SGD. Its 43 paper-facing results expose the most recognizable analytical steps; 35 additional theorem declarations provide algebraic support and recovery variants. The exact replay checks all 78 theorem declarations together with 31 registered real symbols. This narrower framing turns a potential overclaim into a useful trust surface: readers can see precisely which implications are checked, which assumptions are supplied, and which semantic layers remain to be formalized.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used Cursor and its integrated AI models (Claude and GPT families) in order to assist with manuscript drafting, structural editing, and language polishing. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

1. Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*, 177–186. https://doi.org/10.1007/978-3-7908-2604-3_16.
2. Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, 315–323.
3. Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers.
4. Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5), 1–17. [https://doi.org/10.1016/0041-5553\(64\)90137-5](https://doi.org/10.1016/0041-5553(64)90137-5).
5. Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4), 838–855. <https://doi.org/10.1137/0330046>.

6. Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3), 400–407. <https://doi.org/10.1214/aoms/1177729586>.
7. Li, C., Wang, Z., He, W., Wu, Y., Xu, S., & Wen, Z. (2024). Formalization of complexity analysis of the first-order algorithms for convex optimization. arXiv:2403.11437. <https://doi.org/10.48550/arXiv.2403.11437>.
8. Vajjha, K., Trager, B., Shinnar, A., & Pestun, V. (2022). Formalization of a stochastic approximation theorem. In *13th International Conference on Interactive Theorem Proving (ITP 2022)*, LIPIcs 237, Article 31, 31:1–31:18. <https://doi.org/10.4230/LIPIcs.ITP.2022.31>.
9. Bryant, D. (2026). Unconstrained optimization. *Archive of Formal Proofs*. https://isa-afp.org/entries/Unconstrained_Optimization.html.
10. Cassie, B. (2026). *Sgd-lean: Formal Proofs of Bounded-Noise SGD Convergence in Lean 4*. Zenodo. <https://doi.org/10.5281/zenodo.20475582>.

Appendix A: Formal Verification Details

A.1 Proof Architecture

The authoritative companion source is `sgd_proof.py`. A replay on 20 July 2026 checked:

- 78 theorem declarations;
- 31 registered real symbols;
- 109 total registered items, all accepted: the preceding 31 symbols plus 78 theorems;
- zero domain axioms introduced by this field.

The phrase “zero domain axioms” does not mean “unconditional”: theorem-local premises encode every sign, ordering, convexity-shaped, smoothness-shaped, noise, and variance-bound condition.

The trust layers are distinct. First, `verify_all()` replays the declarations in the custom proof environment and checks that the submitted proof steps close the encoded real-arithmetic statements under the bootstrapped arithmetic rules and each theorem’s local premises. The trusted computing base for that result includes the Python runtime and the custom expression, elaboration, tactic, kernel, and real-arithmetic bootstrap implementation. The field has no second independent checker for this replay.

Second, the canonical pipeline exported the current proof source to the companion `sgd__sgd_proof.lean`, neutralized incompatible generated preamble declarations, and compiled the resulting standalone artifact with Lean 4.28.0. The sealed file has zero live sorry tokens. A transitive `#print axioms` audit checked 151 exported theorems and found zero sorryAx dependencies; all 151 depend only on asserted declarations expected under the exporter’s `trusted_as_axioms` boundary. This is genuine independent Lean elaboration of the generated artifact, but it is not an axiom-free reconstruction and does not establish any stronger SGD semantics than the scalar statements encode.

Third, ordinary mathematical and peer review remains responsible for matters outside syntactic proof replay: whether the scalar premises faithfully model the intended optimization setting, whether the paper notation is an appropriate substitution for the source scalars, whether assumptions and references are complete, and whether the selected elementary lemmas are relevant or useful. Machine replay does not decide those semantic or scholarly questions.

A.2 Paper-Facing Theorem Index and Semantic Grades

The grades classify relative roles inside this corpus, not mathematical novelty or depth. **Substantive** is a corpus-relative label for a recognizable analytical implication with a nontrivial rearrangement; **derived** means a useful but immediate consequence of stronger supplied premises; **structural** means domain-neutral sign, product, identity, or rate-shape algebra; **direct-restatement** means the conclusion repeats all or part of a premise without adding mathematical content. Even the 11 corpus-relative substantive rows are short consequences of supplied scalar inequalities.

Theorem Name	Section	Grade	Description
descent_step	§2	derived	Product non-negativity
loss_decreases	§2	substantive	One-step loss ordering
smaller_lr_smaller_step	§2	derived	Learning-rate product ordering
zero_grad_stationary	§2	structural	Zero-product identity
lr_grad_product_nonneg	§2	structural	Product sign
suboptimality_from_convexity	§3	substantive	Convexity-shaped rearrangement
nonneg_suboptimality	§3	derived	Non-negative gap
scaled_suboptimality	§3	derived	Scaled supplied inequality
double_scaled_suboptimality	§3	derived	Double-scaled supplied inequality
strongly_convex_implies_convex	§4	substantive	Drop non-negative quadratic term
strong_convex_quadratic_grow	§4	substantive	Quadratic-growth rearrangement
strong_convex_tighter_suboptimality	§4	substantive	Strong-convexity-shaped rearrangement
gradient_step_descent	§5	substantive	Combine supplied descent bounds
stochastic_step_upper_bound	§5	substantive	Combine supplied stochastic-step bounds
second_moment_upper_bound	§5	substantive	Moment-decomposition consequence
noise_term_nonneg	§5	structural	Noise-product sign
suboptimality_from_distance	§6	substantive	Distance-bound rearrangement
drop_final_distance	§6	substantive	Drop non-negative final term
contraction_preserves_bound	§6	derived	Weaken supplied contraction bound
gradient_dominates_gap	§6	derived	Weaken supplied domination bound
condition_number_ge_one	§6	direct-restatement	Repeats L premise
momentum_accumulation_nonneg	§7	structural	Scalar sum sign
momentum_bounded_by_sum	§7	derived	Scalar momentum upper bound

Theorem Name	Section	Grade	Description
heavy_ball_step_nonneg	§7	structural	Positive product
minibatch_variance_reduction	§8	derived	Positive reciprocal comparison
larger_batch_less_noise	§8	derived	Reciprocal batch comparison
batch_gradient_nonneg	§8	structural	Scalar average sign
svrg_variance_bound	§8	derived	Consequence of assumed variance ordering
control_variate_reduces_variance	§8	derived	Consequence of assumed reduction bound
sublinear_rate_decreases	§9	derived	Reciprocal rate-shape comparison
linear_rate_contraction	§9	structural	One-factor contraction shape
contraction_iterated_shrinks	§9	derived	Two-factor contraction shape
lr_decay_monotone	§10	derived	Reciprocal schedule comparison
warmup_lr_bounded	§10	derived	Linear warmup bound
warmup_lr_nonneg	§10	structural	Warmup expression sign
polyak_average_convex_combination	§11	direct-restatement	Expression equals itself
averaging_weight_decreases	§11	derived	Reciprocal-weight comparison
linear_scaling_rule	§11	structural	Positive scalar multiplication
sqrt_scaling_intermediate	§11	derived	Intermediate scalar ordering
biased_gradient_error_decomposition	§12	substantive	Error-decomposition consequences
unbiased_estimator_mse_equals_variance	§12	derived	Zero-bias substitution
proximal_step_nonneg	§12	structural	Regularization-product sign
regularization_shrinks_norm	§12	derived	Scalar shrinkage inequality

Grade totals: **11 substantive, 20 derived, 10 structural, and 2 direct-restatement**, for 43 paper-facing results.

A.3 Verification Command

After extracting the formal companion bundle, the following command replays the authoritative field when run from the directory containing `sgd_proof.py`:

```
python3 sgd_proof.py
```

Expected output: `s_g_d: verify 109/109 OK, 0 errors`