Latent Optimization

Dr. Tamás Nagy

Dr. Tamás Nagy

tamas@thelatent.space

Skeleton

Abstract

Latent Optimization — ProofEnv proofs.

This paper presents 136 machine-verified theorems building on 1 established facts and 1 hypotheses. All results are formally verified in the Platonic proof kernel (384 verification units, 144 proved statements) and exportable to Lean 4.

---

1. Introduction

2. Further Results

Theorem (rho_opt_gt_one). *Rho Opt Gt One.* [Platonic: rho_opt_gt_one, domain: latent_optimization]

Theorem (rho_opt_pos). *Rho Opt Pos.* [Platonic: rho_opt_pos, domain: latent_optimization]

Theorem (kap_minus_one_pos). *Kap Minus One Pos.* [Platonic: kap_minus_one_pos, domain: latent_optimization]

Theorem (rho_minus_one_identity). *Rho Minus One Identity.* [Platonic: rho_minus_one_identity, domain: latent_optimization]

Theorem (rho_opt_monotone_decreasing). *Rho Opt Monotone Decreasing.* [Platonic: rho_opt_monotone_decreasing, domain: latent_optimization]

Theorem (condition_two_gives_rho_two). *Condition Two Gives Rho Two.* [Platonic: condition_two_gives_rho_two, domain: latent_optimization]

Theorem (kap_rho_duality). *Kap Rho Duality.* [Platonic: kap_rho_duality, domain: latent_optimization]

Theorem (gd_rate_in_unit). *Gd Rate In Unit.* [Platonic: gd_rate_in_unit, domain: latent_optimization]

Theorem (rate_times_rho_is_one). *Rate Times Rho Is One.* [Platonic: rate_times_rho_is_one, domain: latent_optimization]

Theorem (gd_contraction_as_latent_division). *Gd Contraction As Latent Division.* [Platonic: gd_contraction_as_latent_division, domain: latent_optimization]

Theorem (step_count_latent_formula). *Step Count Latent Formula.* [Platonic: step_count_latent_formula, domain: latent_optimization]

Theorem (gradient_analytic). *Gradient Analytic.* [Platonic: gradient_analytic, domain: latent_optimization]

Theorem (gradient_latent_no_larger). *Gradient Latent No Larger.* [Platonic: gradient_latent_no_larger, domain: latent_optimization]

Theorem (total_error_split). *Total Error Split.* [Platonic: total_error_split, domain: latent_optimization]

Theorem (total_error_halving). *Total Error Halving.* [Platonic: total_error_halving, domain: latent_optimization]

Theorem (contraction_shrinks_trunc). *Contraction Shrinks Trunc.* [Platonic: contraction_shrinks_trunc, domain: latent_optimization]

Theorem (contraction_shrinks_opt). *Contraction Shrinks Opt.* [Platonic: contraction_shrinks_opt, domain: latent_optimization]

Theorem (optimal_rate_better). *Optimal Rate Better.* [Platonic: optimal_rate_better, domain: latent_optimization]

Theorem (optimal_rate_in_unit). *Optimal Rate In Unit.* [Platonic: optimal_rate_in_unit, domain: latent_optimization]

Theorem (optimal_rho_gt_naive_rho). *Optimal Rho Gt Naive Rho.* [Platonic: optimal_rho_gt_naive_rho, domain: latent_optimization]

Theorem (optimal_rate_times_rho2_is_one). *Optimal Rate Times Rho2 Is One.* [Platonic: optimal_rate_times_rho2_is_one, domain: latent_optimization]

Theorem (quadratic_fvalue_rho_squared). *Quadratic Fvalue Rho Squared.* [Platonic: quadratic_fvalue_rho_squared, domain: latent_optimization]

Theorem (barrier_nonneg). *Barrier Nonneg.* [Platonic: barrier_nonneg, domain: latent_optimization]

Theorem (suboptimality_gap). *Suboptimality Gap.* [Platonic: suboptimality_gap, domain: latent_optimization]

Theorem (escape_requires_noise). *Escape Requires Noise.* [Platonic: escape_requires_noise, domain: latent_optimization]

Theorem (total_cost_decomposition). *Total Cost Decomposition.* [Platonic: total_cost_decomposition, domain: latent_optimization]

Theorem (grade3_escape_force). *Grade3 Escape Force.* [Platonic: grade3_escape_force, domain: latent_optimization]

Theorem (convex_zero_barrier). *Convex Zero Barrier.* [Platonic: convex_zero_barrier, domain: latent_optimization]

Theorem (nesterov_rho_gt_one). *Nesterov Rho Gt One.* [Platonic: nesterov_rho_gt_one, domain: latent_optimization]

Theorem (nesterov_rate_times_rho_is_one). *Nesterov Rate Times Rho Is One.* [Platonic: nesterov_rate_times_rho_is_one, domain: latent_optimization]

Theorem (nesterov_rate_lt_optimal). *Nesterov Rate Lt Optimal.* [Platonic: nesterov_rate_lt_optimal, domain: latent_optimization]

Theorem (nesterov_rho_gt_optimal). *Nesterov Rho Gt Optimal.* [Platonic: nesterov_rho_gt_optimal, domain: latent_optimization]

Theorem (nesterov_rho_gt_naive). *Nesterov Rho Gt Naive.* [Platonic: nesterov_rho_gt_naive, domain: latent_optimization]

Theorem (nesterov_rho_minus_one_identity). *Nesterov Rho Minus One Identity.* [Platonic: nesterov_rho_minus_one_identity, domain: latent_optimization]

Theorem (newton_rho_exceeds_two). *Newton Rho Exceeds Two.* [Platonic: newton_rho_exceeds_two, domain: latent_optimization]

Theorem (better_hessian_easier_newton). *Better Hessian Easier Newton.* [Platonic: better_hessian_easier_newton, domain: latent_optimization]

Theorem (rho_hierarchy). *Rho Hierarchy.* [Platonic: rho_hierarchy, domain: latent_optimization]

Theorem (newton_eventually_wins). *Newton Eventually Wins.* [Platonic: newton_eventually_wins, domain: latent_optimization]

Theorem (noise_inflates_condition). *Noise Inflates Condition.* [Platonic: noise_inflates_condition, domain: latent_optimization]

Theorem (sgd_rho_le_gd). *Sgd Rho Le Gd.* [Platonic: sgd_rho_le_gd, domain: latent_optimization]

Theorem (zero_noise_recovers_gd). *Zero Noise Recovers Gd.* [Platonic: zero_noise_recovers_gd, domain: latent_optimization]

Theorem (batch_reduces_noise). *Batch Reduces Noise.* [Platonic: batch_reduces_noise, domain: latent_optimization]

Theorem (scaling_law_monotone). *Scaling Law Monotone.* [Platonic: scaling_law_monotone, domain: latent_optimization]

Theorem (intrinsic_le_ambient). *Intrinsic Le Ambient.* [Platonic: intrinsic_le_ambient, domain: latent_optimization]

Theorem (chinchilla_balance). *Chinchilla Balance.* [Platonic: chinchilla_balance, domain: latent_optimization]

Theorem (overparameterization_improves_rho). *Overparameterization Improves Rho.* [Platonic: overparameterization_improves_rho, domain: latent_optimization]

Theorem (preconditioner_improves_rho). *Preconditioner Improves Rho.* [Platonic: preconditioner_improves_rh
domain: latent_optimization]

Theorem (adam_effective_kappa_sqrt). *Adam Effective Kappa Sqrt.* [Platonic: adam_effective_kappa_sqrt, domain: latent_optimization]

Theorem (adam_rho_equals_nesterov). *Adam Rho Equals Nesterov.* [Platonic: adam_rho_equals_nesterov, domain: latent_optimization]

Theorem (adam_beats_uniform_heterogeneous). *Adam Beats Uniform Heterogeneous.* [Platonic: adam_beats_uniform_heterogeneous, domain: latent_optimization]

Theorem (ema_latent_number). *Ema Latent Number.* [Platonic: ema_latent_number, domain: latent_optimization]

Theorem (amsgrad_rho_nondecreasing). *Amsgrad Rho Nondecreasing.* [Platonic: amsgrad_rho_nondecreasing, domain: latent_optimization]

Theorem (bias_correction_restores_kappa). *Bias Correction Restores Kappa.* [Platonic: bias_correction_restores_kappa, domain: latent_optimization]

Theorem (halley_rho_exceeds_newton). *Halley Rho Exceeds Newton.* [Platonic: halley_rho_exceeds_newton, domain: latent_optimization]

Theorem (latent_predicts_grade3_rho). *Latent Predicts Grade3 Rho.* [Platonic: latent_predicts_grade3_rho, domain: latent_optimization]

Theorem (phase_decomposition). *Phase Decomposition.* [Platonic: phase_decomposition, domain: latent_optimization]

Theorem (high_noise_ratio_lt_one). *High Noise Ratio Lt One.* [Platonic: high_noise_ratio_lt_one, domain: latent_optimization]

Theorem (low_noise_slow_exploration). *Low Noise Slow Exploration.* [Platonic: low_noise_slow_exploration, domain: latent_optimization]

Theorem (critical_noise_at_barrier). *Critical Noise At Barrier.* [Platonic: critical_noise_at_barrier, domain: latent_optimization]

Theorem (grokking_is_delayed_transition). *Grokking Is Delayed Transition.* [Platonic: grokking_is_delayed_transition, domain: latent_optimization]

Theorem (grokking_min_delay_grows). *Grokking Min Delay Grows.* [Platonic: grokking_min_delay_grows, domain: latent_optimization]

Theorem (exploitation_uses_local_rho). *Exploitation Uses Local Rho.* [Platonic: exploitation_uses_local_rho, domain: latent_optimization]

Theorem (equal_kappa_equal_rho). *Equal Kappa Equal Rho.* [Platonic: equal_kappa_equal_rho, domain: latent_optimization]

Theorem (rate_rho_inverse). *Rate Rho Inverse.* [Platonic: rate_rho_inverse, domain: latent_optimization]

Theorem (grade1_ceiling_is_nesterov). *Grade1 Ceiling Is Nesterov.* [Platonic: grade1_ceiling_is_nesterov, domain: latent_optimization]

Theorem (nesterov_achieves_ceiling). *Nesterov Achieves Ceiling.* [Platonic: nesterov_achieves_ceiling, domain: latent_optimization]

Theorem (grade2_exceeds_grade1_ceiling). *Grade2 Exceeds Grade1 Ceiling.* [Platonic: grade2_exceeds_grade1_ceiling, domain: latent_optimization]

Theorem (cg_equals_nesterov). *Cg Equals Nesterov.* [Platonic: cg_equals_nesterov, domain: latent_optimization]

Theorem (cg_is_optimal). *Cg Is Optimal.* [Platonic: cg_is_optimal, domain: latent_optimization]

Theorem (preconditioned_cg_improves). *Preconditioned Cg Improves.* [Platonic: preconditioned_cg_improves, domain: latent_optimization]

Theorem (svrg_beats_sgd). *Svrg Beats Sgd.* [Platonic: svrg_beats_sgd, domain: latent_optimization]

Theorem (svrg_matches_gd_large_n). *Svrg Matches Gd Large N.* [Platonic: svrg_matches_gd_large_n, domain: latent_optimization]

Theorem (svrg_hierarchy_position). *Svrg Hierarchy Position.* [Platonic: svrg_hierarchy_position, domain: latent_optimization]

Theorem (two_phase_total_progress). *Two Phase Total Progress.* [Platonic: two_phase_total_progress, domain: latent_optimization]

Theorem (annealing_improves_late_phase). *Annealing Improves Late Phase.* [Platonic: annealing_improves_late_phase, domain: latent_optimization]

Theorem (warmup_suboptimal_start). *Warmup Suboptimal Start.* [Platonic: warmup_suboptimal_start, domain: latent_optimization]

Theorem (discrete_rho_from_flow). *Discrete Rho From Flow.* [Platonic: discrete_rho_from_flow, domain: latent_optimization]

Theorem (larger_step_larger_rho). *Larger Step Larger Rho.* [Platonic: larger_step_larger_rho, domain: latent_optimization]

Theorem (incomplete_preconditioning). *Incomplete Preconditioning.* [Platonic: incomplete_preconditioning, domain: latent_optimization]

Theorem (near_perfect_preconditioner). *Near Perfect Preconditioner.* [Platonic: near_perfect_preconditioner, domain: latent_optimization]

Theorem (precond_strict_improves_rho). *Precond Strict Improves Rho.* [Platonic: precond_strict_improves_rho, domain: latent_optimization]

Theorem (better_precond_lower_kappa). *Better Precond Lower Kappa.* [Platonic: better_precond_lower_kappa, domain: latent_optimization]

Theorem (momentum_is_inverse_rho). *Momentum Is Inverse Rho.* [Platonic: momentum_is_inverse_rho, domain: latent_optimization]

Theorem (momentum_in_unit_interval). *Momentum In Unit Interval.* [Platonic: momentum_in_unit_interval domain: latent_optimization]

Theorem (heavy_ball_momentum_squared). *Heavy Ball Momentum Squared.* [Platonic: heavy_ball_momentum_squared, domain: latent_optimization]

Theorem (worse_condition_more_momentum). *Worse Condition More Momentum.* [Platonic: worse_condition_more_momentum, domain: latent_optimization]

Theorem (per_round_improvement). *Per Round Improvement.* [Platonic: per_round_improvement, domain: latent_optimization]

Theorem (cumulative_improvement). *Cumulative Improvement.* [Platonic: cumulative_improvement, domain: latent_optimization]

Theorem (higher_rho_fewer_steps). *Higher Rho Fewer Steps.* [Platonic: higher_rho_fewer_steps, domain: latent_optimization]

Theorem (regret_decomposition). *Regret Decomposition.* [Platonic: regret_decomposition, domain: latent_optimization]

Theorem (krylov_per_step_contraction). *Krylov Per Step Contraction.* [Platonic: krylov_per_step_contractio domain: latent_optimization]

Theorem (krylov_rho_equals_nesterov). *Krylov Rho Equals Nesterov.* [Platonic: krylov_rho_equals_nesterov, domain: latent_optimization]

Theorem (lanczos_same_rho_as_cg). *Lanczos Same Rho As Cg.* [Platonic: lanczos_same_rho_as_cg, domain: latent_optimization]

Theorem (contraction_implies_latent). *Contraction Implies Latent.* [Platonic: contraction_implies_latent, domain: latent_optimization]

Theorem (step_count_formula). *Step Count Formula.* [Platonic: step_count_formula, domain: latent_optimization]

Theorem (universality_rate_rho_duality). *Universality Rate Rho Duality.* [Platonic: universality_rate_rho_duality, domain: latent_optimization]

Theorem (bernoulli_step_sufficiency). *Bernoulli Step Sufficiency.* [Platonic: bernoulli_step_sufficiency, domain: latent_optimization]

Theorem (bernoulli_monotone_steps). *Bernoulli Monotone Steps.* [Platonic: bernoulli_monotone_steps, domain: latent_optimization]

Theorem (bernoulli_exponential_gap). *Bernoulli Exponential Gap.* [Platonic: bernoulli_exponential_gap, domain: latent_optimization]

Theorem (amgm_sq_sum_dominates). *Amgm Sq Sum Dominates.* [Platonic: amgm_sq_sum_dominates, domain: latent_optimization]

Theorem (cauchy_schwarz_gradient_direction). *Cauchy Schwarz Gradient Direction.* [Platonic: cauchy_schwarz_gradient_direction, domain: latent_optimization]

Theorem (cs_equality_implies_proportional). *Cs Equality Implies Proportional.* [Platonic: cs_equality_implies_proportional, domain: latent_optimization]

Theorem (log_step_count_positive). *Log Step Count Positive.* [Platonic: log_step_count_positive, domain: latent_optimization]

Theorem (higher_rho_fewer_log_steps). *Higher Rho Fewer Log Steps.* [Platonic: higher_rho_fewer_log_steps, domain: latent_optimization]

Theorem (more_precision_more_log_steps). *More Precision More Log Steps.* [Platonic: more_precision_more_log_steps, domain: latent_optimization]

Theorem (opt_sum_nonneg). *Opt Sum Nonneg.* [Platonic: opt_sum_nonneg, domain: latent_optimization]

Theorem (opt_sum_monotone). *Opt Sum Monotone.* [Platonic: opt_sum_monotone, domain: latent_optimization]

Theorem (opt_const_sum). *Opt Const Sum.* [Platonic: opt_const_sum, domain: latent_optimization]

Theorem (opt_convex_combo). *Opt Convex Combo.* [Platonic: opt_convex_combo, domain: latent_optimization]

Theorem (opt_first_order_convexity). *Opt First Order Convexity.* [Platonic: opt_first_order_convexity, domain: latent_optimization]

Theorem (opt_strong_convexity). *Opt Strong Convexity.* [Platonic: opt_strong_convexity, domain: latent_optimization]

Theorem (opt_condition_to_rate). *Opt Condition To Rate.* [Platonic: opt_condition_to_rate, domain: latent_optimization]

Theorem (opt_E_scaling). *Opt E Scaling.* [Platonic: opt_E_scaling, domain: latent_optimization]

Theorem (opt_var_nonneg). *Opt Var Nonneg.* [Platonic: opt_var_nonneg, domain: latent_optimization]

Theorem (opt_var_scaling). *Opt Var Scaling.* [Platonic: opt_var_scaling, domain: latent_optimization]

Theorem (opt_sgd_latent_contraction). *Opt Sgd Latent Contraction.* [Platonic: opt_sgd_latent_contraction, domain: latent_optimization]

3. Convergence Results

Theorem (newton_convergence_condition). *Newton Convergence Condition.* [Platonic: newton_convergence_condition, domain: latent_optimization]

Theorem (ema_geometric_convergence). *Ema Geometric Convergence.* [Platonic: ema_geometric_convergence, domain: latent_optimization]

Theorem (halley_cubic_convergence). *Halley Cubic Convergence.* [Platonic: halley_cubic_convergence, domain: latent_optimization]

Theorem (total_convergence_additive). *Total Convergence Additive.* [Platonic: total_convergence_additive, domain: latent_optimization]

Theorem (exploration_costs_convergence). *Exploration Costs Convergence.* [Platonic: exploration_costs_convergence, domain: latent_optimization]

Theorem (continuous_convergence_rate). *Continuous Convergence Rate.* [Platonic: continuous_convergence_rate, domain: latent_optimization]

Theorem (stability_limit). *Stability Limit.* [Platonic: stability_limit , domain: latent_optimization]

Theorem (bernoulli_convergence_positive). *Bernoulli Convergence Positive.* [Platonic: bernoulli_convergence_positive, domain: latent_optimization]

4. Bounds and Estimates

Theorem (master_inequality). *Master Inequality.* [Platonic: master_inequality, domain: latent_optimization]

Theorem (higher_grade_tighter_bound). *Higher Grade Tighter Bound.* [Platonic: higher_grade_tighter_bound domain: latent_optimization]

Theorem (grade_hierarchy_bounds). *Grade Hierarchy Bounds.* [Platonic: grade_hierarchy_bounds, domain: latent_optimization]

Theorem (phase_cost_lower_bound). *Phase Cost Lower Bound.* [Platonic: phase_cost_lower_bound, domain: latent_optimization]

Theorem (young_noise_step_bound). *Young Noise Step Bound.* [Platonic: young_noise_step_bound, domain: latent_optimization]

Theorem (steepest_descent_cs_bound). *Steepest Descent Cs Bound.* [Platonic: steepest_descent_cs_bound, domain: latent_optimization]

Theorem (opt_total_cost_bound). *Opt Total Cost Bound.* [Platonic: opt_total_cost_bound, domain: latent_optimization]

5. Spectral Theory

Theorem (condition_from_eigenvalues). *Condition From Eigenvalues.* [Platonic: condition_from_eigenvalues, domain: latent_optimization]

Theorem (eigenvalue_spread_rho). *Eigenvalue Spread Rho.* [Platonic: eigenvalue_spread_rho, domain: latent_optimization]

Theorem (eigenvalue_amgm). *Eigenvalue Amgm.* [Platonic: eigenvalue_amgm, domain: latent_optimization]

6. Stability Results

Theorem (saddle_instability_magnitude). *Saddle Instability Magnitude.* [Platonic: saddle_instability_magnitud domain: latent_optimization]

7. Main Theorems

Theorem (grade3_domain_contains_grade2). *Grade3 Domain Contains Grade2.* [Platonic: grade3_domain_contains_grade2, domain: latent_optimization]

8. Formal Framework

Hypotheses

- log_pos_of_gt_one: Log Pos Of Gt One

Established Facts

- log_one_eq_zero: Log One Eq Zero

9. Proof Architecture

All proofs are implemented in the Platonic kernel (elysium/fields/latent_optimization/).

File	Role
latent_optimization_proof.py	

10. Discussion

References