Grigory Evko

GitHub: *github.com/GrigoryEvko* Email: evko.grigory@gmail.com / grigory@theartisan.ai
ORCID: *0000-0001-8566-6204* Phone: +7 918 966 84 55

Research Interests

ML compilers and runtimes (kernel optimization, graph compilation, formal verification), programming language theory (graded dependent types, algebraic effects, session types), and heterogeneous computer architecture (neuromorphic, analog-digital, clockless, adiabatic). Experience in model training (diffusion, super-resolution, autoregressive), GPU inference optimization and production Kubernetes infrastructure at scale.

Research & Professional Experience

- **TheArtisan.ai** Global / 40+ countries
  CTO & Co-Founder *Dec 2023 – Present*
  - Building a B2B SaaS platform delivering AI-personalized images in 40+ countries, serving 1M+ users yearly.
  - Lead a team of developers and manage the technical side end to end: architecture, hiring, planning, code review, hands-on customer service for B2B web platform, API service, physical kiosk software, multi-region GPU infrastructure, and the model training and inference stack.
  - Designed and trained Artisan Upscaler: a 110M-param super-resolution transformer (modified DAT), 14-loss perceptual+GAN objective, NVFP4 quantization-aware training, trained on Blackwell GPUs over a 10M-image corpus.
  - Trained diffusion model on 200K oil paintings – ArtisanXL; designed custom model adapters for identity-preserving portrait generation.
  - Implemented elastic multi-region AWS GPU k8s cluster with inference optimizations for high load model serving with custom inference backends.

- **Institute of Gene Biology, RAS** Moscow, RU
  Bioinformatician *Oct 2021 – Jul 2023*
  - Designed and implemented the computational pipeline for RedC (RNA-DNA proximity ligation) data processing across three prokaryotic species, co-authoring the study with A.A. Gavrilov (***Comm Bio***, 2023).
  - Pipeline subsequently adapted for enhanced Micro-C at 10 bp resolution, contributing to the discovery of novel chromosomal structures in *E. coli* (***Nature***, 2025).
  - Led interdisciplinary bioinformatics workshops across research teams.

- **Independent Research** Remote
  Quantitative Finance & DeFi *Sep 2022 – Nov 2023*
  - Developed time-series prediction model achieving 59% accuracy for short-term price forecasts.
  - Trained EfficientZero (MCTS+RL) model via distributed RL on a 64-GPU cluster.
  - Designed Solidity smart contracts with Python backtesting suite; analyzed liquidity pools protocols mechanics.

Publications

[1] **Elementary 3D organization of active and silenced *E. coli* genome**
A.A. Gavrilov, I. Shamovsky, I. Zhegalova, S. Proshkin, Y. Shamovsky, **G. Evko**, V. Epshtein, A. Rasouly, A. Blavatnik, S. Lahiri, E. Rothenberg, S.V. Razin, E. Nudler
Nature, 2025, vol. 645, pp. 1060–1070. DOI: 10.1038/s41586-025-09396-y
Contributed bioinformatics pipeline adapted for enhanced Micro-C at 10 bp resolution; performed genome annotations and data processing.

[2] **RNA-DNA interactomes of three prokaryotes uncovered by proximity ligation**
A.A. Gavrilov, **G.S. Evko**, A.A. Galitsyna, S.V. Ulianov, T.V. Kochetkova, A.Y. Merkel, A.V. Tyakht, S.V. Razin
Communications Biology (Nature Portfolio), 2023, vol. 6, 473. DOI: 10.1038/s42003-023-04853-8
Co-author. Designed and implemented the complete computational pipeline for RedC (RNA-DNA proximity ligation) data analysis across three prokaryotic species (*E. coli*, *B. subtilis*, *T. adornatum*).

[3] **Di(pyridin-2-yl)amino-substituted 1,10-phenanthrolines and their Ru(II)–Pd(II) dinuclear complexes**
V.A. Ionova, A.V. Dmitrieva, A.S. Abel, **G.S. Evko** et al. ***Dalton Transactions***, 2024, vol. 53, pp. 17021–17035. DOI: 10.1039/D4DT02067G. Synthesized a novel ditopic ligand and its $Ru(bpy)_2$ complex (84% yield).

Education

- **Lomonosov Moscow State University** Moscow, RU
  Faculty of Fundamental Medicine, MD *2020 – 2026*

- **AESC MSU (СУНЦ МГУ)** Moscow, RU
  Prize winner, All-Russian Chemistry Olympiad (ВсОШ) *2018 – 2020*

Research Projects

- **FX – Mechanized Metatheory Kernel for Higher-Dimensional Type Theory** | *Lean 4, FX* *GitHub*
  - Dependent type theory; Lean 4 kernel as a finitely-presented higher-dimensional polygraph: syntax, reduction, coherence as cells across dimensions. Some of metatheory machine-proved, axiom-free: fundamental theorem, strong normalization on closed and open terms, canonicity, consistency, subject reduction, initiality. Global confluence of a non-terminating dependent rewrite system, no normalization assumption ⇒ decidable conversion.
  - Definitional univalence: $\mathrm{Id}_{\mathcal{U}} \rightsquigarrow$ Equiv reduces in-kernel with proven canonicity; univalence, funext and structure-identity discharged by normalization. Type safety and consistency parametric over any ordered grade semiring, binding graded by a vector over a product of resource semirings (usage, effects, security, lifetimes, cost, precision). Transpension, one right adjoint, recovers $\flat/\sharp$ cohesion, guarded $\triangleright$, nominal freshness, Gel/Glue – univalence, internal parametricity and nominal type theory as one construction. Engine fibered over a mode ω-category specializes into cohesive, guarded, linear/graded, parametric and cubical theories; higher and quotient inductive types, definitional path induction, conversion decided by computation.

- **Crucible – Adaptive ML Runtime** | *C++26, FX* *GitHub*
  - Graded-modal type theory unified under compile-time C++26 primitive $\mathsf{Graded}\langle M, L, T\rangle$, lattice laws enforced by a `consteval` concept; linear, refined, tagged, secret, monotonic, fractional-permission and effect-row types are aliases of it. Memory-safety, ownership, information-flow and protocol violations typecheckable and statically verified with GCC 16.
  - Compile-time session types: binary and multiparty, duality $P \leftrightarrow P^{\perp}$, subtyping, crash-stop, global-type projection. Separation-logic permissions threaded through protocol state ($*$-separation, frame rule) enforce permission balance and leak freedom statically.
  - Content-addressed Merkle-DAG record-and-replay below the frontend framework (e.g. Pytorch) dispatch boundary; an interned symbolic shape/stride algebra canonicalizes operations so identical sub-computations share a hash and reuse compiled kernels. Forge optimizes and schedules the DAG into a hardware-agnostic kernel IR, Mimic emits optimized compiled per-vendor ISA for direct interoperability across supported accelerators; sweep-line memory planning renders out-of-memory unrepresentable; kernels federate by content hash across machines.
  - Native eBPF probes on scheduler, lock and syscall tracepoints with hardware PMU counters form an always-live, parallel whole-stack sensing layer, read from userspace through memory-mapped maps with zero syscalls. Lock-free concurrency with primitives (SPSC/MPSC/MPMC/work-stealing) selected at compile time from type-level workload; epidemic-gossip membership over a lock-free hybrid logical clock; bit-reproducible runs with pinned counter-based RNG .

- **Crucible-notes – Multi-Vendor Accelerator Compiler RE** | *C, C++, Python, Lean 4, IDA Pro* *GitHub*
  - Reverse-engineered largest open corpus of proprietary AI compiler/runtime stacks internals stripped binaries: NVIDIA CUDA (cicc, ptxas, nvlink, cudafe++, Tile-IR assembler) and the RT-core, Google TPU, AWS Trainium/Neuron. Recovered per-architecture SASS instruction sets and encoding and latency tables.
  - Fully disassembled ptxas: phase pipeline, IR, resolved bit-by-bit instruction-attribute masks, internal SASS container, RT-core traversal ISA and BVH format. Silicon-validated SASS timing model, Pascal through Blackwell; an SMT scheduler outschedules the vendor's production assembler, bit-identical under a correctness gate, provably never slower; bit-exact semantics machine-checked in Lean 4. Trainium GPSIMD fingerprinted to a commodity DSP core, ISA validated against the vendor's own simulator.

- **Functional Effectful Unit (FEU) – Analog-Digital Processor** | *SPICE, OpenVAF, COMSOL, Python*
  - 3D in-memory analog processor. Charge-redistribution MAC – $Q = C\,V$ across TiN/HfO_2/TiN junctions, Kirchhoff column summation, no transistor in the datapath. IGZO thin-film planes reconfigure the crossbar; balanced-ternary asynchronous (NCL/MTNCL/PCHB) VLIW core. Lithographic IGZO skip-planes form a multiscale PDE preconditioner at zero transistor cost; collision mode makes each junction a trilinear $q\,k\,w$ attention multiplier.
  - Verilog-A device models (charge-conserving IGZO TFT, Fowler–Nordheim MIM, FBAR, skin-effect Cu) via OpenVAF/OSDI; gdstk layout with a custom BEOL DRC deck; SPICE regression, process corners, transistor-level BSIM-BULK Monte-Carlo mismatch. Paper architecture, validated in simulation, awaiting manufacturing technologies progress.

- **go-brrr – Verified Polyglot Program Analysis** | *FX, Rust* *GitHub*
  - Translates source code into a universal IR that infers graded type information (usage, effects, security labels, session types, ownership) from FX's type system for languages that lack these dimensions. Translation correctness proved via CompCert-style simulation relations.
  - Cross-language analysis detects taint propagation across FFI boundaries, ownership violations across language transitions, and information flow leaks through serialization – violations structurally invisible to single-language tools.

Languages

Russian (native), English (C1 proficient)