Generated: 2025-11-08 06:57 UTC

ColorLang — Phase 2 Research Plan

Date: November 7, 2025

Purpose: Define a falsifiable, reproducible Phase 2 program to validate or refine ColorLang's claims through rigorous benchmarks, integrity advances, and scale testing.

Objectives

Validate compression advantage on diverse corpora vs. strong baselines.

Improve robustness via integrity checks and error-correcting encoding.

Characterize performance at scale; explore GPU-backed execution/rendering.

Workstreams

1) Build corpus and benchmarks (WS1)

2) Integrity + ECC layer (WS2)

3) Scale tests + GPU prototype (WS3)

WS1 — Corpus and Benchmarks

Corpus tiers:

Tier A (Low entropy): uniform blocks, palettes, tile repetition.

Tier B (Structured): gradients, sparse instruction clusters, typical UI grids.

Tier C (High entropy): pseudo-random color noise controls.

Baselines:

Generic: PNG, WebP-lossless, PNG+zstd, WebP+zstd.

ColorLang: palette, RLE, hybrid; pattern re-introduced with stringified keys.

Protocol:

Compute size (bytes), ratio r = Sc/So, savings (1–r).

Decode speed: ms to reconstruct image/program; VM parse+decode throughput (pix/ms).

Optional: full run time for small programs (VM cycles/s).

Reproducibility:

Fix Python version and libs; pin seeds for synthetic corpora.

Emit bench_results.json per run with metadata (env, commit, params).

WS2 — Integrity + ECC Layer

Region checksums: per tile (e.g., 16×16) SHA-256; optional Merkle over tiles.

ECC palettes: assign hue bands with parity bits or redundant channels.

Robust decode:

Tolerate ±δ hue perturbations; majority vote across redundant pixels.

Validate opcode-class band before operand extraction.

Security hardening: strict bounds, structured exceptions, fuzz target.

WS3 — Scale Tests + GPU Prototype

Scale matrix: grid sizes {512^2, 1024^2, 2048^2, 4096^2}.

Metrics: render time, parse time, memory footprint, throughput (pix/ms).

GPU path (prototype):

Use a GPU texture to store HSV grid; shader/kernel to map pixels→opcodes.

CPU compare: ensure semantic equivalence for a subset.

Goal: feasibility study, not production-ready speed.

Metrics and Success Criteria

Compression advantage: Median savings of Hybrid vs. PNG+zstd ≥ 5% across Tier A/B; allow Tier C parity.

Decode performance: ColorLang decode + parse within ±10% of PNG decode on Tier A/B.

Robustness: Misdecode rate < 1e-6 under hue noise $\delta \leq 2$ units using ECC.

Scale viability: 2048^2 grids render+parse in < 250 ms on reference machine.

Risks and Mitigations

Synthetic bias → Include Tier C controls; report per-tier.

Timing noise → Use high-resolution timers; repeat runs; report variance.

Serialization limits → Stringify pattern keys; consider binary container v2.

GPU complexity → Limit to shader prototype and equivalence tests.

Deliverables

D1: bench/ harness with dataset generator, codec runners, and report emitter.

D2: Integrity/ECC module integrated into parser/VM; docs and tests.

D3: Scale benchmark report with CPU vs. GPU proto comparison.

D4: Updated thesis appendix with Phase 2 results and discussion.

Environment and Repro Steps (planned)

```
pwsh

python -m venv .venv

.\.venv\Scripts\Activate.ps1

python -m pip install -U pip wheel

pip install -r requirements.txt

python bench\run_benchmarks.py
```

Timeline (indicative)

Week 1–2: WS1 corpus + baseline harness; initial results.

Week 3: WS2 integrity/ECC, fuzzing, robustness assessment.

Week 4: WS3 scale tests, GPU prototype, consolidate report.

Decision Gates

If median savings < 5% vs. PNG+zstd, reposition claims to “competitive” rather than superior.

If robustness targets fail, prioritize ECC redesign before scale.

If GPU path shows no advantage, deprioritize for Phase 3.

Appendices

A: Metric definitions and formulas.

B: Proposed JSON schema for bench_results.json.

C: Threats to validity alignment with Challenger Review.