

Stage■11 / Step■1 Launch Packet

Bridging from consolidated benchmark results into live integration tests.

1) Background

Stage■10 established the geodesic parser (residual energies, matched filter, dual gates). Stage■11 reframes this as Warp → Detect → Denoise. On Latent■ARC (n=100), the denoiser path achieved 100/100 exact with hallucination ≈ 0.5% and omission ≈ 0.2%, surpassing both stock and Stage■10 baselines. Phantom index (≈0.065) and margin (≈0.044) confirm a single dominant cognition well.

2) Objective

Transition from planning/benchmarking into Stage■11 / Step■1 live execution. Verify that Warp + Detect + Denoise maintain single■well dominance and deterministic convergence when applied in integration contexts.

3) Prerequisites

- 1 Benchmark script (`stage11-benchmark-latest.py`) available and tested.
- 2 Calibration run completed: phantom index ≤0.07, margin ≥0.04.
- 3 Latent■ARC reference results (denoiser path) saved: accuracy_exact=1.0, hallucination≈0.005.
- 4 Plots of warped PCA manifold and fitted funnel rendered.

4) Command Set (Minimal)

Shadow (Warp only)

```
python3 stage11-benchmark-latest.py \ --samples 100 --seed 42 \ --render_well \ --out_csv stage11_metrics_shadow.csv \ --out_json stage11_summary_shadow.json
```

Detect (Warp + parser)

```
python3 stage11-benchmark-latest.py \ --samples 100 --seed 42 \ --use_funnel_prior 1 \ --alpha 0.05 --beta_s 0.25 --q_s 2 \ --tau_rel 0.60 --tau_abs_q 0.93 --null_K 40 \ --out_csv stage11_metrics_detect.csv \ --out_json stage11_summary_detect.json
```

Denoise (full Stage■11)

```
python3 stage11-benchmark-latest.py \ --samples 100 --seed 42 \ --latent_arc --latent_dim 64 --latent_arc_noise 0.05 \ --denoise_mode hybrid --ema_decay 0.85 --median_k 3 \ --probe_k 5 --probe_eps 0.02 \ --conf_gate 0.65 --noise_floor 0.03 \ --seed_jitter 2 \ --out_csv latent_arc_denoise_100.csv \ --out_json latent_arc_denoise_100.json
```

5) Success Criteria

Metric	Target
Phantom Index	≤ 0.07 (observed ~0.065)
Margin	≥ 0.04 (observed ~0.044)

Accuracy (denoise)	1.0 (100/100 exact)
Hallucination (denoise)	≈ 0.005 (noise floor)
Omission (denoise)	≈ 0.002
Precision / Recall	$\approx 0.998 / 0.999$

6) Next Actions

- 1 Freeze golden config (CLI, seed, plots, JSON/CSV).
- 2 Patch visualization sort() warning (replace key=... with np.argsort or sorted).
- 3 Expand Step■1 testing to additional slices and stress scenarios (scaling walls, hidden targets).
- 4 Prepare for Step■2 (LLM integration): layer tap scan, shadow mode logging, confidence■gated rescoring.