

Stage-11 → Text LLM Integration Roadmap (10 Steps)

1. Confirm the Doctrine

Warp → Detect → Denoise (Stage-11) is the core doctrine. Always-on warp ($\alpha_{\min} > 0$) with soft gating is required for convergence. This principle holds across embeddings and must carry into text models.

2. Reproduce Stock Baseline

Implement a `text_arc_stock_baseline.py` using GPT-2 small. Run ARC-style prompts in greedy decode (rudimentary) to get baseline performance. This mirrors embedding ARC tests, giving an apples-to-apples baseline.

3. Apply ARC Testbed Equivalence

Ensure same ARC framework is applied to text as was to embeddings. This controls for test conditions so gains are attributable to warp/denoise, not benchmark mismatch.

4. Introduce Warp Layer Hook

Insert forward hook at layer -9. Perform PCA-2 slice to define radius + inward vector. Apply small inward curvature per token (α_{\min}).

5. Add Soft Trend Gate

Replace hard thresholds with sigmoids on trend (k_{tr} , τ). Latch + linger logic to stabilize bursts.

6. Integrate Detect (Gain Only)

Matched filter (window, sigma) + null calibration. Multiply trend gain; detect never acts as permission. Detect refines warp amplification but warp always remains active.

7. Incorporate Denoiser

Implement soft denoiser from `stage11_ab_eval_base_denoise.py`: EMA + median smoothing, phantom guard, jitter averaging. Ensure denoiser only scales/smooths, never flips sign.

8. Telemetry & Metrics

Track α sequence, radius shrink, burst length, applied rate. Compare against embedding metrics for convergence.

9. CPU/GPU Profiles

Provide CPU-friendly runs for quick checks. GPU configs for budget T4/L4 (v4b) and richer A100 runs.

10. Consolidate into Unified Script

Merge stock + geo decode modes (`--gen_mode stock|geo`) as in A/B eval base. One script runs both, using a config JSON or CLI args. Ensures all R&D; (embeddings → text) is properly carried forward.