Stage■11 LLM Integration Plan — From Latent Doctrine to Layer■9 Wiring

Context

Stage■11 (Warp → Detect → Denoise) has already been captured in patents and supporting math documents, demonstrating deterministic convergence on latent embeddings. These filings explicitly describe latent benchmarks on hidden states, thereby establishing the doctrine as prior art. What remains is to extend this foundation into live LLM integration (e.g., terraform layer 9 during decode).

What is Already Covered

- Doctrine: Warp → Detect → Denoise is patented, with math, metrics, and latent benchmarks.
- Filings specify latent embeddings (hidden states), which covers interventions inside network layers.
- Hidden states = layer embeddings, so conceptually the filings already include layer wiring.

What is New (and May or May Not Need a Separate Filing)

- Hook engineering (forward hook, PCA projection, clamped nudge) is implementation detail, not doctrine.
- Portable calibration profiles, runtime guards, APIs → packaging/software engineering, not novel core IP.
- Broader concept: Stage■11 as a pluggable LLM tool or universal runtime module may justify a continuation claim.

Rule of Thumb

• If the goal is just protection for terraform@■9 in GPT■2 (or similar), existing patents suffice.
• If the goal is to claim Stage■11 as a universal LLM plug■in framework (profiles, runtime guards, telemetry), then a continuation or divisional filing makes sense.

Suggested Next Steps

- Decide whether protection is needed only for doctrine (already filed) or also for a framework claim.
- If only doctrine: no new filing needed, proceed with integration experiments.
- If framework claim: prepare continuation language focused on pluggable integration of Stage■11 into arbitrary LLMs.

Summary

The doctrine filings already cover layer■level interventions conceptually. No new patent is required just for wiring to layer 9. However, a continuation could be filed if the goal is to claim Stage■11 as a general■purpose plug■in framework for LLMs.