DeFi Project – Milestones 6–12 (Detailed Plan)

Scope: updated Tier 1/2 trajectory — trained mapper on synthetic latents now; WDD and real latents deferred to later milestones.

Milestone 6 — Train & Wire the Mapper (Prompt → Synthetic Latent)

Objective

- Replace hashmap lookups with a trained lightweight mapper that encodes prompt text to synthetic latent vectors used by Stage■10/11 rails.
- Keep rail logic unchanged; only the prompt→latent leg becomes learned.

Inputs

- Curated prompt set (≥1k diverse prompts across primitives: deposit/withdraw/borrow/repay/swap/{add,remove}_collateral).
- Target synthetic latent generator (same dimensionality and semantics used in Tier 0).
- Policies: ltv_max, near_margin, mapper.confidence_threshold.

Work Items

- Model: start with logistic■regression or small MLP; optional TF■IDF/BPE features. Produce fixed■dim latent.
- Training script: `milestones/defi_milestone6.py --train --data --out .artifacts/defi_mapper.joblib`.
- Calibration: sweep `confidence_threshold ∈ {0.6,0.7,0.8}`; export ROC/PR to `.artifacts/`.
- Plumbing: update `defi_mapper.py` to load model_path; expose `predict(prompt) -> (latent, conf)`.

CLI / Example

```
`python3 milestones/defi_milestone6.py --train data/defi_mapper_train.jsonl --out
.artifacts/defi_mapper.joblib`
```

Pass/Exit Criteria

- Top■1 primitive agreement vs Tier■0 ≥ 95% on a 20% holdout.
- Mapper emits `conf` and abstains when `conf < threshold` (coverage reported).

Artifacts

- .artifacts/defi_mapper.joblib, .artifacts/defi_mapper_calibration.json, curves: ROC/PR PNGs.

Milestone 7 — Parser + Matched Filter Bench (Stage■11 lite)

Objective

- Validate end■to■end with the trained mapper in the loop using synthetic latents, without WDD.
- Quantify stability (top■1 consistency), abstain behavior, and guard compatibility.

Inputs

- Suite: `benchmarks/suites/defi_dist_v2.jsonl` (2–5k prompts; balanced primitives + edge phrasing).
- Policy grid: confidence_threshold ∈ {0.6,0.7,0.8}, near_margin ∈ {0.85,0.90,0.95}.
- Context: oracle freshness window (age_sec, max_age_sec).

Work Items

- Bench runner: `benchmarks/defi/bench_driver.py --suite … --runs k --policy … --out …`
- Emit JSON + CSV + Markdown report; include per■primitive confusion and abstain counts.
- Add MICROLLM_DEBUG prints at verify boundaries (reason tokens: ltv, hf, oracle, abstain_non_exec).

Metrics

- Top■1 accuracy, stability@runs, abstain rate, policy■block rate, guard precision/recall.
- Latency per prompt (p50/p95).

Pass/Exit Criteria

- Top■1 ≥ 92% overall; abstain ≤ 8%; zero false■negative guard escapes on edge suites.

Milestone 8 — Scale Bench to 2–5k and Export Dashboards

Objective

- Run the full distribution test at scale and produce artifacts for comparative analysis.

Work Items

- Grid search over thresholds (thr × near_margin) with fixed runs (e.g., 3).
- Aggregate results into one parquet; compute micro/macro averages per primitive.
- Generate comparison plots (accuracy vs. abstain, ROC■like operating curve).

CLI / Example

```
for thr in 0.6 0.7 0.8; do for m in 0.85 0.90 0.95; do python3
benchmarks/defi/bench_driver.py --suite benchmarks/suites/defi_dist_v2.jsonl \ --rails
stage11 --runs 3 --context '{"oracle":{"age_sec":5,"max_age_sec":30}}' \ --policy
"{\"ltv_max\":0.75, \"near_margin\":$m, \"mapper\":{\"model_path\":\".artifacts/defi_map
per.joblib\",\"confidence_threshold\":$thr}}" \ --out
.artifacts/dist_v2_thr${thr}_m${m}.json; done; done
```

Pass/Exit Criteria

- Operating point selected that maximizes accuracy subject to abstain ≤ 10% and zero guard escapes.

Milestone 9 — Guard Edge Suites (LTV / HF / Oracle)

Objective

- Construct and run adversarial edge suites to verify deterministic guard firing and clear reason tokens.

Work Items

- Create suites: `defi_edges_ltv.jsonl`, `defi_edges_hf.jsonl`, `defi_edges_oracle.jsonl`.
- Ensure per■case expectations: top1=None with reason in {ltv, hf, oracle} when violating policy.
- Bench command emits per■case traces and verify summaries (.md).

Pass/Exit Criteria

- 100% correct reason tokens on policy■blocked cases; no false approvals.

Milestone 10 — Introduce WDD on Synthetic Latents

Objective

- Add Warp→Detect→Denoise around synthetic traces; measure delta vs. stock rails.

Work Items

- Enable rails.use_wdd=True and rails.denoise=True paths behind a policy flag.
- Implement slim detector (energy/threshold) and a median/EMA denoiser; export `report.denoised`.
- Ablation: compare stock vs. WDD at same operating point (thr, near_margin).

Metrics

- ∆ accuracy (+), ∆ abstain (–/neutral), stability@runs↑; verify reasons unchanged on blocked cases.

Pass/Exit Criteria

- WDD yields measurable improvement (≥+1.5pp accuracy) without increasing guard escapes.

Milestone 11 — Sidecar Real Latents (Prototype)

Objective

- Replace synthetic generator with sidecar LLM latents; keep the mapper and rails intact.

Work Items

- Define latent schema & normalization to match existing rails.
- Collect prompt→latent pairs (small n=1–5k) and re■calibrate mapper if needed.
- Run the same benches; document shift between synthetic vs. real latents.

Pass/Exit Criteria

- Parit y within –3pp accuracy vs. synthetic baseline with equivalent abstain; guards still deterministic.

Milestone 12 — Full Tier■2 Benchmark & Docs

Objective

- Consolidate all improvements and publish a comprehensive report with reproducible scripts.

Work Items

- Single script to run full grid on real latents + WDD and export JSON/CSV/plots.
- Writeup with methodology, datasets, policy settings, and failure analysis.
- Prepare release notes and README section with exact CLI incantations.

Pass/Exit Criteria

- Green runs across distribution + edges; documented operating point; downloadable artifacts in .artifacts/.