AELP / AELP2 — Technical Overview

Generated 2025-09-29 12:53 UTC

1) What problem we solve

We predict and assemble winning ad creatives and portfolios before spending, using offline data from Meta and vendor sources. The system ingests historical performance, scores creative blueprints with a trained new■ad ranker, calibrates CPM/CTR/CVR baselines by placement, simulates CAC/volume via Monte■Carlo, then packages a day■by■day launch plan with per■ad setup.

2) High■level architecture

```
Data & Vendors → Ingestion → Feature/Model → Forecasts/RL → Planner UI → Launch
[Meta Ads API] [Vendor CSV/SearchAPI] \ / \ / v v +-----------------------+ |
AELP2/pipelines & | | AELP2/tools importers | +-----------------------+ | v
+-----------------------+ | Creative objects & | | features (reports/) |
+-----------------------+ | v +-----------------------+ | New■ad ranker + | |
baselines/forecasts | +-----------------------+ | | | v | Offline bandit sim v
+-----------------------+ | Creative Planner UI | | (Next API + Vite UI) |
+-----------------------+ | v +-----------------------+ | Setup Playbooks / | |
Launch & Monitoring | +-----------------------+
```

Key environment (masked)

```
export GOOGLE_CLOUD_PROJECT=aura-t…
export BIGQUERY_USERS_DATASET=***
export BIGQUERY_TRAINING_DATASET=gaelp_…
export BIGQUERY_DATASET=gaelp_…
export GA4_PROPERTY_ID=proper…
export NODE_ENV=***
export GCS_BUCKET=gaelp-…
export CREATIVE_GCS_BUCKET=gaelp-…
export REDIS_HOST=***
export REDIS_PORT=***
export REDIS_URL=redis:…
export OPENAI_API_KEY=sk-svc…
export ANTHROPIC_API_KEY=sk-ant…
export META_ACCESS_TOKEN=EAAZBi…
export META_ADLIBRARY_ACCESS_TOKEN=EAAZBi…
export META_API_VERSION=***
export META_ADLIBRARY_COUNTRIES=***
export META_APP_ID=440073…
export META_APP_SECRET=***
export META_BUSINESS_ID=477239…
export META_ACCOUNT_ID=act_19…
export META_PRIMARY_ACTION=***
export GOOGLE_ADS_DEVELOPER_TOKEN=uikJ5k…
export GOOGLE_ADS_CLIENT_ID=637142…
export GOOGLE_ADS_CLIENT_SECRET=GOCSPX…
export GOOGLE_ADS_REFRESH_TOKEN=1//04T…
export GOOGLE_ADS_LOGIN_CUSTOMER_ID=***
export GOOGLE_ADS_CUSTOMER_ID=***
export TIKTOK_ACCESS_TOKEN=your_t…
export TIKTOK_APP_ID=your_t…
```

3) Pipelines & Tools

```
1) Meta → BigQuery (by placement): meta_to_bq.py --by_placement →
meta_ad_performance_by_place 2) Vendor/Ad Library imports: fetch_searchapi_meta.py
→ vendor_imports/*.csv → import_vendor_meta_creatives.py 3) Features + Scoring:
build_features_from_creative_objects.py → creative_features.jsonl
score_vendor_creatives.py → vendor_scores.json (p_win, lcb) 4) Baselines &
Forecasts: compute_us_paid_baselines_by_place.py → us_meta_baselines_by_place.json
forecast_us_cac_volume.py → us_cac_volume_forecasts.json (+ Balance) 5) RL Prep
and Simulation: add_novelty_and_export_rl_pack.py → rl_test_pack.json,
asset_briefs.json simulate_bandit_from_forecasts.py → rl_offline_simulation.json
6) UI/API (Planner): Next app routes → /api/planner/* (forecasts, RL, vendor
scores) Vite UI → /creative-planner (Top■K, packages, setup)
```

Key scripts

AELP2/pipelines/ads_common.py

AELP2/pipelines/ads_mcc_coordinator.py

AELP2/pipelines/attribution_engine_stub.py

AELP2/pipelines/audience_expansion.py

AELP2/pipelines/auto_recalibration.py

AELP2/pipelines/bandits_writer.py

AELP2/pipelines/bid_edit_proposals.py

AELP2/pipelines/bid_landscape_modeling.py

AELP2/pipelines/bing_to_bq.py

AELP2/pipelines/build_affiliate_triggered.py

AELP2/pipelines/build_ga_aligned_daily.py

AELP2/pipelines/build_impact_partner_domains.py

AELP2/pipelines/build_triggered_series.py

AELP2/pipelines/calibration_parallel_stub.py

AELP2/pipelines/calibration_stratified_views.py

AELP2/pipelines/canary_monitoring.py

AELP2/pipelines/canary_timeline_writer.py

AELP2/pipelines/channel_attribution_r.py

AELP2/pipelines/channel_attribution_runner.py

AELP2/pipelines/check_data_quality.py

AELP2/pipelines/competitive_intel_ingest.py

AELP2/pipelines/copy_optimizer_stub.py

AELP2/pipelines/cost_monitoring_stub.py

AELP2/pipelines/create_bq_views.py

AELP2/pipelines/create_channel_views.py

AELP2/pipelines/creative_ab_planner.py

AELP2/pipelines/creative_bandit_head.py

AELP2/pipelines/creative_embeddings_stub.py

AELP2/pipelines/creative_fatigue_alerts.py

AELP2/pipelines/cross_platform_kpi_daily.py

AELP2/pipelines/dayparting_optimizer.py

AELP2/pipelines/delayed_conversions_stub.py

AELP2/pipelines/fidelity_evaluation.py

AELP2/pipelines/fidelity_parallel_stub.py

AELP2/pipelines/ga4_backfill_audience_breakdown.py

AELP2/pipelines/ga4_build_attribution.py

AELP2/pipelines/ga4_build_audience_breakdown.py

AELP2/pipelines/ga4_build_audiences.py

AELP2/pipelines/ga4_build_derived.py

AELP2/pipelines/ga4_extract_affiliate_clickids.py

AELP2/pipelines/ga4_lagged_attribution.py

AELP2/pipelines/ga4_paths_attribution.py

AELP2/pipelines/ga4_permissions_check.py

AELP2/pipelines/ga4_to_bq.py

AELP2/pipelines/generate_qs_alerts.py

AELP2/pipelines/generate_qs_fix_tickets.py

AELP2/pipelines/google_ads_ad_performance_to_bq.py

AELP2/pipelines/google_ads_adgroups_to_bq.py

AELP2/pipelines/google_ads_assets_to_bq.py

AELP2/pipelines/google_ads_conversion_actions_to_bq.py

AELP2/pipelines/google_ads_conversion_stats_by_action_to_bq.py

AELP2/pipelines/google_ads_discover_accounts.py

AELP2/pipelines/google_ads_geo_device_to_bq.py

AELP2/pipelines/google_ads_keywords_to_bq.py

AELP2/pipelines/google_ads_mcc_to_bq.py

AELP2/pipelines/google_ads_search_terms_to_bq.py

AELP2/pipelines/google_ads_to_bq.py

AELP2/pipelines/google_recommendations_scanner.py

AELP2/pipelines/gsc_to_bq.py

AELP2/pipelines/hints_to_proposals.py

AELP2/pipelines/impact_backfill_performance.py

AELP2/pipelines/impact_clicks_to_bq.py

AELP2/pipelines/impact_entities_to_bq.py

AELP2/pipelines/impact_run_report_to_bq.py

AELP2/pipelines/impact_to_bq.py

AELP2/pipelines/journey_path_summary.py

AELP2/pipelines/journeys_populate.py

AELP2/pipelines/kpi_consistency_check.py

AELP2/pipelines/kpi_crosscheck.py

AELP2/pipelines/linkedin_to_bq.py

AELP2/pipelines/load_affiliate_ach_costs.py

AELP2/pipelines/load_promo_calendar.py

AELP2/pipelines/lp_ab_hooks_stub.py

AELP2/pipelines/ltv_priors.py

AELP2/pipelines/meta_to_bq.py

AELP2/pipelines/mmm_attributed_service.py

AELP2/pipelines/mmm_lightweightmmm.py

AELP2/pipelines/mmm_service.py

AELP2/pipelines/model_registry_stub.py

AELP2/pipelines/module_runner.py

AELP2/pipelines/namespace_refactor_report.py

AELP2/pipelines/offpolicy_eval.py

AELP2/pipelines/opportunity_outcomes.py

AELP2/pipelines/opportunity_scanner.py

AELP2/pipelines/ops_alerts_stub.py

AELP2/pipelines/parity_report.py

AELP2/pipelines/permissions_check.py

AELP2/pipelines/platform_skeleton_log.py

AELP2/pipelines/policy_hints_writer.py

AELP2/pipelines/portfolio_optimizer.py

AELP2/pipelines/privacy_audit_stub.py

AELP2/pipelines/propensity_uplift.py

AELP2/pipelines/quality_signal_daily.py

AELP2/pipelines/quality_softgate_stub.py

AELP2/pipelines/realtime_budget_pacer.py

AELP2/pipelines/reconcile_posthoc.py

AELP2/pipelines/rl_policy_hints_writer.py

AELP2/pipelines/robyn_runner.py

AELP2/pipelines/robyn_validator.py

AELP2/pipelines/rule_engine.py

AELP2/pipelines/security_audit.py

AELP2/pipelines/segments_to_audiences.py

AELP2/pipelines/slo_watch_stub.py

AELP2/pipelines/tiktok_to_bq.py

AELP2/pipelines/training_posthoc_reconciliation.py

AELP2/pipelines/trust_gates_evaluator.py

AELP2/pipelines/uplift_eval.py

AELP2/pipelines/upload_google_offline_conversions.py

AELP2/pipelines/upload_meta_capi_conversions.py

AELP2/pipelines/users_db_stub.py

AELP2/pipelines/value_bridge.py

AELP2/pipelines/youtube_reach_planner.py

Tools

AELP2/tools/ad_level_accuracy.py

AELP2/tools/ad_level_calibration_v22.py

AELP2/tools/ad_level_calibration_v23.py

AELP2/tools/ad_level_ranker_v24.py

AELP2/tools/add_novelty_and_export_rl_pack.py

AELP2/tools/add_phone_look.py

AELP2/tools/annotate_weekly_with_policy.py

AELP2/tools/assemble_boring_good.py

AELP2/tools/assemble_enforced.py

AELP2/tools/assemble_from_assets.py

AELP2/tools/assemble_original_batch.py

AELP2/tools/assemble_pattern.py

AELP2/tools/assemble_real_ad.py

AELP2/tools/assemble_spot_the_tell.py

AELP2/tools/assemble_two_screens.py

AELP2/tools/assemble_with_vo.py

AELP2/tools/attention_heuristics.py

AELP2/tools/backfill_ad_daily_insights.py

AELP2/tools/bigspy_auto_export.py

AELP2/tools/build_brand_pack.py

AELP2/tools/build_cool_variant.py

AELP2/tools/build_features_from_creative_objects.py

AELP2/tools/build_labels_weekly.py

AELP2/tools/build_placement_calibrators.py

AELP2/tools/build_proof_assets.py

AELP2/tools/build_system_overview_pdf.py

AELP2/tools/build_topk_from_scores.py

AELP2/tools/build_weekly_predictions.py

AELP2/tools/cascade_dr_topk.py

AELP2/tools/check_dual_gate_thresholds.py

AELP2/tools/collect_kb_overlays.py

AELP2/tools/compute_aesthetic_metrics.py

AELP2/tools/compute_locked_targets.py

AELP2/tools/compute_mmm_bands.py

AELP2/tools/compute_motion_features.py

AELP2/tools/compute_target_cac_frontier.py

AELP2/tools/compute_us_meta_baselines.py

AELP2/tools/compute_us_paid_baselines_by_place.py

AELP2/tools/conformal_topk.py

AELP2/tools/conformal_topk_weekly.py

AELP2/tools/crawl_youtube_titles.py

AELP2/tools/demo_storyboard.py

AELP2/tools/detect_objects_yolo.py

AELP2/tools/download_runway_results.py

AELP2/tools/dual_gate_weekly.py

AELP2/tools/eleven_design_voice.py

AELP2/tools/enrich_creatives_with_objects.py

AELP2/tools/enrich_weekly_with_cpc.py

AELP2/tools/estimate_policy_uplift.py

AELP2/tools/eval_ablation.py

AELP2/tools/eval_wbua.py

AELP2/tools/eval_wbua_cluster.py

AELP2/tools/eval_wbua_forward.py

AELP2/tools/eval_wbua_novel.py

AELP2/tools/export_meta_ad_configs.py

AELP2/tools/export_meta_creative_outcomes.py

AELP2/tools/export_ui_overlays_stub.py

AELP2/tools/extract_audio_features.py

AELP2/tools/extract_audio_lufs.py

AELP2/tools/extract_keyframes.py

AELP2/tools/extract_legibility_features.py

AELP2/tools/extract_visual_embeddings.py

AELP2/tools/fetch_campaign_placement_conversions.py

AELP2/tools/fetch_meta_adlibrary.py

AELP2/tools/fetch_meta_creatives.py

AELP2/tools/fetch_searchapi_meta.py

AELP2/tools/fidelity_eval_roll.py

AELP2/tools/finalize_gate.py

AELP2/tools/fit_cac_calibrator.py

AELP2/tools/forecast_us_cac_volume.py

AELP2/tools/forward_forecast.py

AELP2/tools/gen_eleven_vo.py

AELP2/tools/gen_eleven_vo_pack.py

AELP2/tools/gen_runway_hooks.py

AELP2/tools/gen_veo_hooks.py

AELP2/tools/generate_advantage_manifest.py

AELP2/tools/generate_balance_blueprints.py

AELP2/tools/generate_blueprints.py

AELP2/tools/generate_score_loop.py

AELP2/tools/hourly_multipliers.py

AELP2/tools/import_proof_clips.py

AELP2/tools/import_vendor_meta_creatives.py

AELP2/tools/join_dna_to_creatives.py

AELP2/tools/join_finals_features.py

AELP2/tools/journey_states_from_bq.py

AELP2/tools/log_creative_dna.py

AELP2/tools/make_meta_account_audit.py

AELP2/tools/merge_copy_banks.py

AELP2/tools/meta_campaign_bayes.py

AELP2/tools/mine_meta_copy.py

AELP2/tools/mj_ingest.py

AELP2/tools/normalize_audio.py

AELP2/tools/offline_creative_search.py

AELP2/tools/offline_rl_stub.py

AELP2/tools/ope_config_weighting.py

AELP2/tools/ope_topk.py

AELP2/tools/ope_upgrades.py

AELP2/tools/ope_uplift_topk.py

AELP2/tools/parse_brand_guide.py

AELP2/tools/policy_audit_live.py

AELP2/tools/portfolio_selector.py

AELP2/tools/propose_kb_non_claims.py

AELP2/tools/pull_google_ads_copy.py

AELP2/tools/pull_google_ads_copy_from_bq.py

AELP2/tools/pull_google_ads_copy_rest.py

AELP2/tools/pull_impact_copy.py

AELP2/tools/pull_impact_copy_from_bq.py

AELP2/tools/qc_gates.py

AELP2/tools/render_bayes_report.py

AELP2/tools/render_quiz_overlay.py

AELP2/tools/render_sim_onepager.py

AELP2/tools/report_policy_vs_mixed.py

AELP2/tools/rl_shadow_score.py

AELP2/tools/score_ad_items.py

AELP2/tools/score_new_ads.py

AELP2/tools/score_vendor_creatives.py

AELP2/tools/selector_utility_baseline.py

AELP2/tools/self_judge.py

AELP2/tools/serve_previews.py

AELP2/tools/sim_fidelity_campaigns.py

AELP2/tools/sim_fidelity_campaigns_journey.py

AELP2/tools/sim_fidelity_campaigns_temporal.py

AELP2/tools/sim_fidelity_campaigns_temporal_v2.py

AELP2/tools/sim_fidelity_campaigns_temporal_v3.py

AELP2/tools/sim_fidelity_eval.py

AELP2/tools/sim_fidelity_eval_empirical.py

AELP2/tools/sim_fidelity_journey_criteo.py

AELP2/tools/simulate_bandit_from_forecasts.py

AELP2/tools/sort_uploaded_assets.py

AELP2/tools/still_to_motion.py

AELP2/tools/tag_ad_patterns.py

AELP2/tools/train_new_ad_ranker.py

AELP2/tools/tune_generator_priors.py

AELP2/tools/uplift_baseline_test.py

AELP2/tools/validate_candidates.py

AELP2/tools/validate_kb.py

AELP2/tools/voice_library_add_prompt.py

AELP2/tools/weekly_fidelity_from_v3.py

AELP2/tools/weekly_relabel.py

AELP2/tools/weekly_topN_dual.py

Apps & APIs

AELP2/apps/dashboard (Next.js API + UI)

AELP2/external/growth-compass-77 (Vite React UI)

Data lineage (producer → artifact → consumer)

vendor_imports/*.csv → import_vendor_meta_creatives.py → reports/creative_objects/

creative_objects/ → build_features_from_creative_objects.py → reports/creative_features/creative_features.jsonl

creative_features.jsonl + models/new_ad_ranker → score_vendor_creatives.py → reports/vendor_scores.json

BigQuery meta_ad_performance_by_place → compute_us_paid_baselines_by_place.py → reports/us_meta_baselines_by_place.json

reports/us_meta_baselines.json + ad_blueprints_top20.json → forecast_us_cac_volume.py → reports/us_cac_volume_forecasts.json

reports/us_balance_forecasts.json (Balance forecasts generator) → reports/rl_balance_pack.json

reports/us_cac_volume_forecasts.json → simulate_bandit_from_forecasts.py → reports/rl_offline_simulation.json

reports/* (forecasts, RL, vendor scores) → Next API /api/planner/* → Vite UI /creative-planner

4) Forecasting & Accuracy

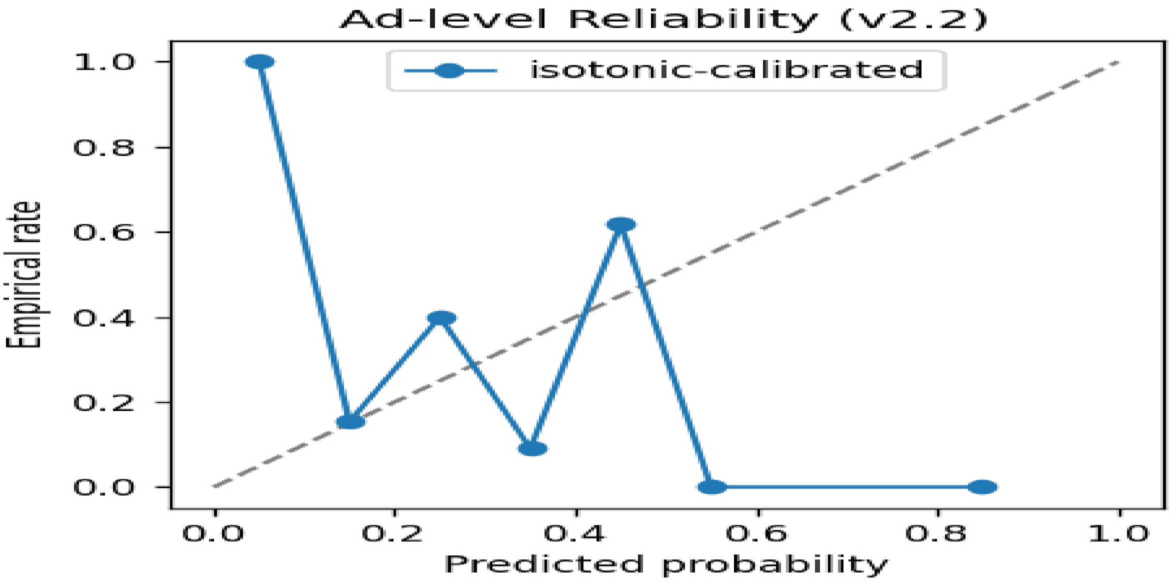
Vendor creatives scored: 1650

Forecast items — Security: 20, Balance: 20

Baseline keys (by placement): 23

Ranker precision@5: 0.26666666666666667, precision@10: 0.3

Calibration (reliability curve):



5) Creative Planner & Launch

The Planner exposes forecasts, RL packs, and setup checklists. It provides package builders for $30k/$50k daily budgets, per■ad instructions (Campaign/Ad Set/Ad), and exports (PDF/ZIP/JSON).

Endpoints

/api/planner/forecasts — security/balance forecasts

/api/planner/vendor-scores — scored vendor creatives

/api/planner/rl — rl packs + offline sim

/api/planner/setup/[creative_id] — per■ad setup checklist

6) Ops & How to Run

Servers: Next API (3000), Vite UI (8080).

Start API: cd AELP2/apps/dashboard && npm run build && PORT=3000 NODE_ENV=production npm run start

Start UI: cd AELP2/external/growth-compass-77 && npm run build && npm run preview -- --host 127.0.0.1 --port 8080

Preview finals: python3 AELP2/tools/serve_previews.py and tunnel 8080

Data refresh (typical sequence)

1) Meta ingest (by placement) → 2) Baselines → 3) Vendor import → 4) Features → 5) Score → 6) Forecasts → 7) RL sim → 8) Planner

8) Risks & Mitigations

- API rate limits → backoff + window slicing in meta_to_bq.py
- US Ad Library gaps → supplement with SearchAPI/vendor CSVs; rely on our own Meta performance for baselines
- Forecast drift → recalibrate baselines weekly; placement-aware CVR clamps; conformal lower bounds

9) Script docstrings (headnotes)

tools/ad_level_accuracy.py: Compute ad-level offline accuracy metrics from historical data and simulator predictions.

tools/ad_level_calibration_v22.py: no docstring

tools/ad_level_calibration_v23.py: no docstring

tools/ad_level_ranker_v24.py: no docstring

tools/add_novelty_and_export_rl_pack.py: no docstring

tools/add_phone_look.py: Apply a subtle smartphone look: grain, slight vignette, gentle motion jitter.

tools/annotate_weekly_with_policy.py: Join adset/campaign config into weekly creatives and mark policy compliance.

tools/assemble_boring_good.py: Assemble 3 "boringly good" Aura variants:

tools/assemble_enforced.py: Assemble a 9:16 ad with enforced slot grammar using a manifest + recipe.

tools/assemble_from_assets.py: no docstring

tools/assemble_original_batch.py: Assemble fully original ads from Runway clips and VO packs.

tools/assemble_pattern.py: Assemble a pattern-driven ad using:

tools/assemble_real_ad.py: Assemble a "real" ad using Runway hooks, a proof clip, a relief clip, captions, and ElevenLabs VO.

tools/assemble_spot_the_tell.py: Assemble a 9:16 "Spot The Tell" video:

tools/assemble_two_screens.py: Assemble a 9:16 split-screen "Two Screens, Two Outcomes" video.

tools/assemble_with_vo.py: Assemble a Veo/Runway hook with ElevenLabs VO, add end-card and disclaimer overlay, normalize audio.

tools/attention_heuristics.py: First-3s attention heuristics (rough):

tools/backfill_ad_daily_insights.py: Backfill ad-level daily insights in small date chunks with checkpointing.

tools/bigspy_auto_export.py: Headless BigSpy exporter (best-effort, cookie-based login reuse)

tools/build_brand_pack.py: Build a minimal brand pack from existing ads + site:

tools/build_cool_variant.py: Build a polished preview ad from your brand + MJ assets without waiting on remote video models.

tools/build_features_from_creative_objects.py: Build lightweight numeric features for creatives using the cached Meta

tools/build_labels_weekly.py: Construct pairwise training data (diff features, label win vs baseline) from

tools/build_placement_calibrators.py: no docstring

tools/build_proof_assets.py: Build simple proof overlay PNGs and an end-card image from brand pack.

tools/build_system_overview_pdf.py: Build a comprehensive PDF (for non-coders) that explains AELP/AELP2:

tools/build_topk_from_scores.py: no docstring

tools/build_weekly_predictions.py: Build weekly (calendar) predictions and actuals per creative from ad-level daily insights.

tools/cascade_dr_topk.py: Cascade-DR style estimator for a Top-K slate policy.

tools/check_dual_gate_thresholds.py: Check dual-gate precision and yield against thresholds and write a status file.

tools/collect_kb_overlays.py: Collect claim overlays (text + mandatory disclaimer) and a default CTA from KB for use in end-cards.

tools/compute_aesthetic_metrics.py: Compute lightweight aesthetic/quality proxies (no-ref):

tools/compute_locked_targets.py: Compute locked target_CAC per campaign from recent weekly files to stabilize gates

tools/compute_mmm_bands.py: Compute simple MMM-like daily spend bands (min/base/max) from recent spend and returns using a concave response proxy.

tools/compute_motion_features.py: Compute simple motion features from finals using OpenCV optical flow.

tools/compute_target_cac_frontier.py: Compute per-campaign target_CAC by scanning historical CAC percentiles and choosing the point that maximizes

tools/compute_us_meta_baselines.py: Compute US Meta baselines from BigQuery and write to reports/us_meta_baselines.json.

tools/compute_us_paid_baselines_by_place.py: Compute US Meta paid-event baselines by publisher_platform and placement from

tools/conformal_topk.py: no docstring

tools/conformal_topk_weekly.py: Selection-conditional conformal bounds for weekly creatives.

tools/crawl_youtube_titles.py: Lightweight crawler to fetch video titles/descriptions from public YouTube channel pages without API keys.

tools/demo_storyboard.py: Generate a 4-frame 9:16 storyboard from the Identity KB:

tools/detect_objects_yolo.py: Detect objects on mid-frames using YOLOv8n (CPU). Reports counts of person and cell phone.

tools/download_runway_results.py: no docstring

tools/dual_gate_weekly.py: Dual-gate weekly evaluator.

tools/eleven_design_voice.py: Design and create an ElevenLabs voice from a natural-language prompt, then store it in our library.

tools/enrich_creatives_with_objects.py: no docstring

tools/enrich_weekly_with_cpc.py: Enrich weekly creatives with a CPC-mix based predicted CAC.

tools/estimate_policy_uplift.py: Estimate coarse uplift scalars for key policy switches by comparing compliant-like vs noncompliant items within campaign-weeks.

tools/eval_ablation.py: no docstring

tools/eval_wbua.py: Evaluate Weekly Baseline Uplift Accuracy (WBUA) on weekly_creatives/* files.

tools/eval_wbua_cluster.py: Cluster-holdout WBUA: treat creative "families" as clusters using ad name

tools/eval_wbua_forward.py: Forward-holdout WBUA: freeze decisions using only history up to week t-1 and

tools/eval_wbua_novel.py: Compute WBUA (Weekly Baseline Uplift Accuracy) for novel-only creatives.

tools/export_meta_ad_configs.py: Export current campaign and adset configuration from Meta Graph API.

tools/export_meta_creative_outcomes.py: Export historical per-ad outcomes from Meta (read-only) and build files for ad-level accuracy.

tools/export_ui_overlays_stub.py: Stub exporter for UI overlays: create timing JSON entries for proof inserts.

tools/extract_audio_features.py: Extract simple audio features via ffprobe:

tools/extract_audio_lufs.py: Extract LUFS using ffmpeg loudnorm analyze mode for finals.

tools/extract_keyframes.py: Extract simple keyframes for local MP4s (finals) to AELP2/reports/keyframes//tXXXX.jpg.

tools/extract_legibility_features.py: Compute legibility proxy per final by reusing self_judge (contrast of top band).

tools/extract_visual_embeddings.py: Compute OpenCLIP embeddings for keyframes under AELP2/reports/keyframes/* and

tools/fetch_campaign_placement_conversions.py: no docstring

tools/fetch_meta_adlibrary.py: Meta Ad Library fetcher

tools/fetch_meta_creatives.py: Fetch Meta Ad and AdCreative objects for historical ads listed in AELP2/reports/creative/*.json.

tools/fetch_searchapi_meta.py: Fetch Meta Ad Library results via SearchAPI.io (self-serve) and write CSVs that our

tools/fidelity_eval_roll.py: Rolling-origin evaluation for the temporal (Phase 2) simulator.

tools/finalize_gate.py: Finalize a candidate video with hard gates so low-quality outputs never publish.

tools/fit_cac_calibrator.py: Fit a monotonic calibrator mapping predicted CAC to actual CAC using isotonic regression.

tools/forecast_us_cac_volume.py: Monte Carlo CAC/volume forecast for Top-20 blueprints at $30k and $50k budgets.

tools/forward_forecast.py: Forward forecast check using Phase 2 (temporal) model.

tools/gen_eleven_vo.py: Generate VO lines with ElevenLabs v3 and save MP3s.

tools/gen_eleven_vo_pack.py: Generate a pack of VO lines from AELP2/prompts/vo_scripts.json using ElevenLabs.

tools/gen_runway_hooks.py: Generate 9:16 hook clips on Runway (Gen■4 Turbo) using image_to_video.

tools/gen_veo_hooks.py: Generate 9:16 hook clips on Vertex AI (Veo 3 Fast) and write to GCS.

tools/generate_advantage_manifest.py: Create a simple Advantage+ Creative manifest from finals.

tools/generate_balance_blueprints.py: Generate Balance-focused creative blueprints from Aura parental-controls concepts

tools/generate_blueprints.py: Generate synthetic creative blueprints (visual + copy/offer/CTA/placement) as

tools/generate_score_loop.py: Orchestrate generate -> feature -> score -> filter loop (stub).

tools/hourly_multipliers.py: Fetch account-level hourly CVR multipliers (normalized to 1.0 mean).

tools/import_proof_clips.py: Import real screen recordings and cut proof clips for slot B.

tools/import_vendor_meta_creatives.py: Normalize vendor exports (BigSpy/PowerAdSpy/SocialPeta) into GAELP creative_objects.

tools/join_dna_to_creatives.py: Join heuristic CreativeDNA-style features to historical ad items using ad_name and placement mix.

tools/join_finals_features.py: Join per-final features into one JSONL for analysis.

tools/journey_states_from_bq.py: Optional: Pull journey states from BigQuery if available.

tools/log_creative_dna.py: Write CreativeDNA + cost placeholders for produced finals; load to BigQuery if configured.

tools/make_meta_account_audit.py: Generate a Macro Account Audit checklist for Meta (offline).

tools/merge_copy_banks.py: Merge copy from Meta (copy_bank.json), Google Ads (google_ads_copy.json), Impact (impact_copy.json), and YouTube (youtube_copy.json).

tools/meta_campaign_bayes.py: Bayesian per-campaign MMM (Poisson with geometric adstock) for Meta campaigns.

tools/mine_meta_copy.py: no docstring

tools/mj_ingest.py: Ingest Midjourney backplates from a folder, normalize to 9:16, lightly denoise,

tools/normalize_audio.py: Normalize audio to -14 LUFS using ffmpeg loudnorm. If no audio, adds a silent track.

tools/offline_creative_search.py: Offline creative generation + simulator scoring scaffold.

tools/offline_rl_stub.py: Offline RL stub (bandit/IQL-lite): trains a logistic policy to predict P(win) from features and applies a conservative penalty.

tools/ope_config_weighting.py: OPE with config-based propensity reweighting (diagnostic).

tools/ope_topk.py: no docstring

tools/ope_upgrades.py: OPE upgrades: DM, IPS, SNIPS, DR, SWITCH, CAB; plus simple bandit-FQE.

tools/ope_uplift_topk.py: no docstring

tools/parse_brand_guide.py: Parse brand guide PDFs to extract a usable brand_config.json for the creative engine.

tools/policy_audit_live.py: Policy audit of current live adsets/campaigns vs META_POLICY_SETUP.md

tools/portfolio_selector.py: Portfolio selector: choose a per-campaign slate that maximizes predicted purchases under CAC caps and test budget share.

tools/propose_kb_non_claims.py: Propose non-claim benefit lines for each product KB from merged copy bank (Meta/Google/Impact/YouTube).

tools/pull_google_ads_copy.py: Pull Google Ads headlines/descriptions via the Google Ads API (read-only) using env credentials.

tools/pull_google_ads_copy_from_bq.py: Pull Google Ads ad names/headlines from BigQuery (preferred over API on this box).

tools/pull_google_ads_copy_rest.py: Pull Google Ads headlines/descriptions via REST (searchStream) using OAuth refresh token.

tools/pull_impact_copy.py: Pull top affiliate (Impact) copy snippets via Impact API.

tools/pull_impact_copy_from_bq.py: Pull affiliate (Impact) copy-like fields from BigQuery tables populated by AELP2 pipelines.

tools/qc_gates.py: Hard QC gates for candidate clips and finals.

tools/render_bayes_report.py: no docstring

tools/render_quiz_overlay.py: Render timed quiz-style overlay cards on a 9:16 video using ffmpeg drawbox/drawtext.

tools/render_sim_onepager.py: Render a simple, marketer-friendly one-pager (Markdown) summarizing simulator fidelity.

tools/report_policy_vs_mixed.py: Compare WBUA and dual-gate metrics on mixed vs policy-annotated data.

tools/rl_shadow_score.py: RL shadow scorer using temporal v2 simulator parameters.

tools/score_ad_items.py: Score ad items using success_config proxies (best-effort on available fields).

tools/score_new_ads.py: Score local final videos with the Meta-only new-ad ranker (light features).

tools/score_vendor_creatives.py: Score all creative_objects (including vendor-imported) with the trained new-ad ranker.

tools/selector_utility_baseline.py: Baseline utility selector: rank creatives by expected utility proxy and export Top-N per campaign (offline only).

tools/self_judge.py: Self-judging evaluator for creative quality.

tools/serve_previews.py: Serve AELP2/outputs/finals on http://127.0.0.1:8080 for local/SSH-tunnel preview.

tools/sim_fidelity_campaigns.py: Phase 1 fidelity: per-campaign heterogeneity

tools/sim_fidelity_campaigns_journey.py: Phase 3 fidelity: journey stages + recency proxy

tools/sim_fidelity_campaigns_temporal.py: Phase 2 fidelity: temporal patterns (weekday seasonality + frequency fatigue)

tools/sim_fidelity_campaigns_temporal_v2.py: Temporal simulator v2: time-decayed CVR priors + auto window (per-campaign)

tools/sim_fidelity_campaigns_temporal_v3.py: Temporal simulator v3: per-campaign hourly effect + fast-drift window rule

tools/sim_fidelity_eval.py: Simulation Fidelity Check (AELP ↔ AELP2)

tools/sim_fidelity_eval_empirical.py: no docstring

tools/sim_fidelity_journey_criteo.py: Journey + Criteo fidelity evaluation (no BigQuery writes).

tools/simulate_bandit_from_forecasts.py: Offline bandit simulation using the forecast distributions for Top-20 blueprints.

tools/sort_uploaded_assets.py: Sort a big "drop" folder of mixed digital assets into the repo structure.

tools/still_to_motion.py: Turn a still image (1080x1920 recommended) into a subtle-motion MP4.

tools/tag_ad_patterns.py: Tag ads with hook_type, emotion, proof_device, captions_present (heuristics on text),

tools/train_new_ad_ranker.py: Train a lightweight pairwise ranker on weekly pairs (feature diffs).

tools/tune_generator_priors.py: no docstring tools/uplift_baseline_test.py: Historic baseline uplift test.

tools/validate_candidates.py: no docstring

tools/validate_kb.py: no docstring

tools/voice_library_add_prompt.py: no docstring

tools/weekly_fidelity_from_v3.py: Compute weekly (7-day aggregated) fidelity from sim_fidelity_campaigns_temporal_v3.json.

tools/weekly_relabel.py: Recompute weekly labels from creative_enriched/*.json and write to creative_weekly/*.json

tools/weekly_topN_dual.py: Export weekly Top-N creatives that pass the dual gate with a conservative lower-bound utility.

pipelines/ads_common.py: Shared utilities for Google Ads → BigQuery loaders.

pipelines/ads_mcc_coordinator.py: MCC Coordinator: Orchestrate Ads data loads across all child accounts.

pipelines/attribution_engine_stub.py: Real Attribution Engine Implementation for AELP2

pipelines/audience_expansion.py: Audience Expansion Tooling (shadow-only).

pipelines/auto_recalibration.py: Auto-recalibration (shadow-only): detect drift and log proposals.

pipelines/bandits_writer.py: Bandits Posteriors Writer (heuristic P0)

pipelines/bid_edit_proposals.py: Bid Edit Proposals (shadow-only, stub).

pipelines/bid_landscape_modeling.py: Bid Landscape Modeling (stub): derive CPC↔volume curves per campaign.

pipelines/bing_to_bq.py: Stub: Bing Ads to BigQuery adapter (pilot)

pipelines/build_affiliate_triggered.py: Affiliate-triggered (delayed-reward) series from GA4 export.

pipelines/build_ga_aligned_daily.py: Build GA-aligned KPI daily table that keeps GA as the majority signal and fills only

pipelines/build_impact_partner_domains.py: Build partner→domain mapping from Impact MediaPartners table.

pipelines/build_triggered_series.py: Build touch-aligned (delayed-reward) daily series from GA4 export.

pipelines/calibration_parallel_stub.py: Parallel calibration probes (stub): demonstrates multiprocessing fanout across probe params.

pipelines/calibration_stratified_views.py: Create calibration stratified views (by channel/device) for RL vs Ads.

pipelines/canary_monitoring.py: Canary Monitoring (shadow): detect anomalies and write ops_alerts.

pipelines/canary_timeline_writer.py: Writes a simple canary timeline to BQ for dashboard consumption.

pipelines/channel_attribution_r.py: ChannelAttribution (R) weekly job — containerized stub with BQ writer.

pipelines/channel_attribution_runner.py: Channel Attribution Runner (skeleton)

pipelines/check_data_quality.py: Basic data quality and freshness checks for core BigQuery tables.

pipelines/competitive_intel_ingest.py: Competitive Intelligence ingest (stub): ensure auction insights table exists.

pipelines/copy_optimizer_stub.py: Copy Optimization Loop (policy-safe, stub).

pipelines/cost_monitoring_stub.py: Cost monitoring (stub): compute daily cost and emit ops_alerts if exceeding cap.

pipelines/create_bq_views.py: Create standard BigQuery views for dashboards/subagents.

pipelines/create_channel_views.py: Create channel-specific daily views from ads_campaign_performance using advertising_channel_type.

pipelines/creative_ab_planner.py: Creative AB Planner (shadow-only): proposes a test and writes to BQ.

pipelines/creative_bandit_head.py: Creative Bandit Head (stub): logs decision proposals to ab_experiments (shadow-only).

pipelines/creative_embeddings_stub.py: Production Creative Embeddings System for AELP2

pipelines/creative_fatigue_alerts.py: Creative Fatigue Detection (stub): flags CTR/CVR decay over recent days.

pipelines/cross_platform_kpi_daily.py: Cross-Platform KPI Daily (stub): aggregates KPI metrics across platforms.

pipelines/dayparting_optimizer.py: Dayparting Optimizer (stub): propose hour/day schedule caps.

pipelines/delayed_conversions_stub.py: Production Delayed Conversions Processing for AELP2

pipelines/fidelity_evaluation.py: Fidelity Evaluation: Compare simulation/RL telemetry vs GA4/Ads (MAPE/RMSE/KS) and write results to BigQuery.

pipelines/fidelity_parallel_stub.py: Parallel fidelity evaluation (stub): shards dates and would call fidelity_evaluation per shard.

pipelines/ga4_backfill_audience_breakdown.py: WITH events AS (

pipelines/ga4_build_attribution.py: Build GA attribution daily tables from export using purchase-level cookies (first pass LND proxy):

pipelines/ga4_build_audience_breakdown.py: Build high-intent breakdown by channel (source/medium) from GA4 export.

pipelines/ga4_build_audiences.py: Build high-intent audiences from GA4 export:

pipelines/ga4_build_derived.py: Build GA-derived daily/monthly KPI tables and cohort summaries from GA4 export (events_*).

pipelines/ga4_extract_affiliate_clickids.py: Extract affiliate click IDs from GA4 export into a joinable table.

pipelines/ga4_lagged_attribution.py: GA4 Lagged Attribution Importer

pipelines/ga4_paths_attribution.py: Build multi-touch attribution from GA4 export:

pipelines/ga4_permissions_check.py: GA4 permissions check (dry-run friendly): verifies Data API access or provides guidance.

pipelines/ga4_to_bq.py: Ingest GA4 aggregates into BigQuery (ga4_aggregates).

pipelines/generate_qs_alerts.py: SELECT DATE(date) AS d, campaign_id, ANY_VALUE(campaign_name) AS name,

pipelines/generate_qs_fix_tickets.py: WITH base AS (

pipelines/google_ads_ad_performance_to_bq.py: Load Google Ads ad performance into BigQuery (ads_ad_performance).

pipelines/google_ads_adgroups_to_bq.py: Load Google Ads ad group performance into BigQuery (ads_ad_group_performance).

pipelines/google_ads_assets_to_bq.py: Google Ads Assets ingestion.

pipelines/google_ads_conversion_actions_to_bq.py: Load Google Ads conversion actions into BigQuery (ads_conversion_actions).

pipelines/google_ads_conversion_stats_by_action_to_bq.py: Load Google Ads conversion stats by conversion_action into BigQuery

pipelines/google_ads_discover_accounts.py: Discover accessible Google Ads accounts (including MCC/manager accounts).

pipelines/google_ads_geo_device_to_bq.py: Load Google Ads device-level performance into BigQuery (ads_geo_device_performance).

pipelines/google_ads_keywords_to_bq.py: Load Google Ads keyword performance into BigQuery (ads_keyword_performance).

pipelines/google_ads_mcc_to_bq.py: Enumerate all child accounts under an MCC and load Ads performance into BigQuery.

pipelines/google_ads_search_terms_to_bq.py: Load Google Ads search terms into BigQuery (ads_search_terms).

pipelines/google_ads_to_bq.py: Ingest Google Ads performance into BigQuery (ads_campaign_performance).

pipelines/google_recommendations_scanner.py: Google Ads Recommendations Scanner (safe, shadow-only).

pipelines/gsc_to_bq.py: Google Search Console → BigQuery (brand vs non-brand trend).

pipelines/hints_to_proposals.py: Promote policy hints to shadow proposals (HITL path, stub).

pipelines/impact_backfill_performance.py: Impact.com daily performance backfill (auto-discovery).

pipelines/impact_clicks_to_bq.py: no docstring

pipelines/impact_entities_to_bq.py: Impact.com entities → BigQuery (MediaPartners, Ads, Deals, Invoices, TrackingValueRequests).

pipelines/impact_run_report_to_bq.py: Try the official Run endpoint first; if forbidden, fall back to ReportExport.

pipelines/impact_to_bq.py: Impact.com (Impact Radius) → BigQuery loader.

pipelines/journey_path_summary.py: Summarize user journey paths and transition probabilities.

pipelines/journeys_populate.py: Journeys Populate (bootstrap)

pipelines/kpi_consistency_check.py: KPI Consistency Check: compares KPI-only view vs training episodes daily metrics.

pipelines/kpi_crosscheck.py: KPI Cross-check: compare ads_kpi_daily view vs aggregated ads_campaign_performance.

pipelines/linkedin_to_bq.py: no docstring

pipelines/load_affiliate_ach_costs.py: Load external affiliate (ACH) payouts into BigQuery to complete partner costs.

pipelines/load_promo_calendar.py: no docstring pipelines/lp_ab_hooks_stub.py: Landing-Page A/B Hooks (stub): propose UTM cohorts and GA4 goal names.

pipelines/ltv_priors.py: LTV Priors Daily: computes simple 30/90-day LTV priors per segment from uplift scores.

pipelines/meta_to_bq.py: Meta → BigQuery loader (schema ensure + ingestion).

pipelines/mmm_attributed_service.py: MMM with Proper Delayed Reward Attribution (3-14 day windows)

pipelines/mmm_lightweightmmm.py: LightweightMMM runner with safe fallbacks and BQ logging.

pipelines/mmm_service.py: MMM v1 (Lightweight, dependency-free):

pipelines/model_registry_stub.py: Production Model Registry for AELP2

pipelines/module_runner.py: LP Module Runner (background job)

pipelines/namespace_refactor_report.py: Namespace Refactor Report (stub): scans repo for non-AELP2 imports and prints summary.

pipelines/offpolicy_eval.py: Off-policy evaluation (stub): compares hints vs realized outcomes.

pipelines/opportunity_outcomes.py: Opportunity Outcomes (stub): summarize approvals and subsequent performance.

pipelines/opportunity_scanner.py: Opportunity Scanner v1 (Google-first, shadow)

pipelines/ops_alerts_stub.py: Ops Alerts (stub): writes a placeholder alert row.

pipelines/parity_report.py: Parity Report (stub): compares ads_kpi_daily vs training_episodes_daily.

pipelines/permissions_check.py: Permissions & Accounts Checker (writes to BQ).

pipelines/platform_skeleton_log.py: no docstring

pipelines/policy_hints_writer.py: Policy Hints writer (stub): writes exploration/budget tilt hints to BQ.

pipelines/portfolio_optimizer.py: Portfolio Optimizer (stub): propose daily cross-campaign allocations under CAC cap.

pipelines/privacy_audit_stub.py: Privacy audit (stub): ensures free-text fields are not stored; writes summary row.

pipelines/propensity_uplift.py: Propensity/Uplift Bootstrap: writes `segment_scores_daily` using simple exposed vs unexposed deltas.

pipelines/quality_signal_daily.py: Quality Signal Daily (stub): computes a simple quality proxy (trial→paid, retention).

pipelines/quality_softgate_stub.py: Quality soft-gate (stub): reads quality_signal_daily and emits safety_events when below threshold.

pipelines/realtime_budget_pacer.py: Real-time budget pacer (stub): emits pacing proposals at minute cadence.

pipelines/reconcile_posthoc.py: Post-hoc reconciliation of RL vs Ads/GA4 metrics by date.

pipelines/rl_policy_hints_writer.py: RL Policy Hints Writer (bootstrap)

pipelines/robyn_runner.py: Robyn Validator Runner (skeleton)

pipelines/robyn_validator.py: Robyn weekly validator (containerized R) — runner stub with BQ summary write.

pipelines/rule_engine.py: Rule Engine (stub): evaluates safe rules and records actions (HITL required).

pipelines/security_audit.py: Security Audit (enhanced stub): records IAM/audit status notes and ADC context.

pipelines/segments_to_audiences.py: Segments → Audiences mapping (shadow-only).

pipelines/slo_watch_stub.py: SLO/Alerting stub: scans ops_flow_runs and safety_events for failures; emits ops_alerts.

pipelines/tiktok_to_bq.py: no docstring

pipelines/training_posthoc_reconciliation.py: Post-hoc reconciliation (lag-aware KPIs).

pipelines/trust_gates_evaluator.py: Trust Gates Evaluator: computes pass/fail for pilot gates and writes to BQ.

pipelines/uplift_eval.py: Uplift v1: Bootstrap segment uplift evaluation from journey tables (if present).

pipelines/upload_google_offline_conversions.py: Google Offline Conversions Upload (HITL-gated):

pipelines/upload_meta_capi_conversions.py: Meta CAPI Conversions Upload (HITL-gated): builds hashed payload and logs intent.

pipelines/users_db_stub.py: Production User Database Management for AELP2

pipelines/value_bridge.py: Value-Based Bidding Bridge (stubs)

pipelines/youtube_reach_planner.py: YouTube Reach Planner — real API when available.

10) What to change safely

- AOV assumptions in forecasting scripts
- Chip sets/filters in AELP2/config/*.yaml for vendor imports
- Number of creatives per package in Planner UI