# ORFS-agent: Tool-Using Agents for Chip Design Optimization

MLCAD 2025

**Amur Ghose, Andrew B. Kahng, Sayak Kundu, Zhiang Wang**
UCSD

September 9 2025

# Roadmap

1. **Motivation**
2. Concept & Contributions
3. Method
4. Experiments
5. Results
6. Limitations & Next
7. Takeaways

Open, reproducible; replication badges.

## Motivation — EDA flow tuning is high-dimensional

- Knobs: *hundreds–thousands*; RTL→GDS

# Motivation — EDA flow tuning is high-dimensional

- Knobs: *hundreds–thousands*; RTL→GDS
- Plain BO: weak domain priors; objective design burden; poor high-D scaling

# Motivation — EDA flow tuning is high-dimensional

- Knobs: *hundreds–thousands*; RTL$\rightarrow$GDS
- Plain BO: weak domain priors; objective design burden; poor high-D scaling
- Agents: read logs; reason in context; call tools; iterate

## Motivation — EDA flow tuning is high-dimensional

- Knobs: *hundreds–thousands*; RTL→GDS
- Plain BO: weak domain priors; objective design burden; poor high-D scaling
- Agents: read logs; reason in context; call tools; iterate
- Core loop: propose → parallel ORFS runs → read metrics/logs → refine

## Motivation — EDA flow tuning is high-dimensional

- Knobs: *hundreds–thousands*; RTL→GDS
- Plain BO: weak domain priors; objective design burden; poor high-D scaling
- Agents: read logs; reason in context; call tools; iterate
- Core loop: propose → parallel ORFS runs → read metrics/logs → refine
- Model-agnostic: no fine-tune; swap in stronger LLMs

# Intuition — semantics-aware search $>$ blind BO

- Classical BO treats the vector of knobs as *opaque*; kernel + acquisition on an unlabeled space scales poorly in high-$D$

# Intuition — semantics-aware search $>$ blind BO

- Classical BO treats the vector of knobs as *opaque*; kernel + acquisition on an unlabeled space scales poorly in high-$D$
- Here, knobs have *names and physics*: e.g., `Core Util` $\uparrow \Rightarrow$ congestion risk; `Clock Period` $\downarrow \Rightarrow$ timing pressure

# Intuition — semantics-aware search $>$ blind BO

- Classical BO treats the vector of knobs as *opaque*; kernel + acquisition on an unlabeled space scales poorly in high-$D$
- Here, knobs have *names and physics*: e.g., `Core Util` $\uparrow \Rightarrow$ congestion risk; `Clock Period` $\downarrow \Rightarrow$ timing pressure
- The agent exploits semantics and logs: recognizes regimes, uses priors (ranges, monotonicities), and proposes *meaningful* batches

# Intuition — semantics-aware search > blind BO

- Classical BO treats the vector of knobs as *opaque*; kernel + acquisition on an unlabeled space scales poorly in high-$D$
- Here, knobs have *names and physics*: e.g., `Core Util` $\uparrow \Rightarrow$ congestion risk; `Clock Period` $\downarrow \Rightarrow$ timing pressure
- The agent exploits semantics and logs: recognizes regimes, uses priors (ranges, monotonicities), and proposes *meaningful* batches
- Loop: read logs $\Rightarrow$ summarize context $\Rightarrow$ choose sampling scheme (local sweep, bracket, or exploratory LHS) $\Rightarrow$ refine

# Intuition — semantics-aware search > blind BO

- Classical BO treats the vector of knobs as *opaque*; kernel + acquisition on an unlabeled space scales poorly in high-$D$
- Here, knobs have *names and physics*: e.g., `Core Util` $\uparrow \Rightarrow$ congestion risk; `Clock Period` $\downarrow \Rightarrow$ timing pressure
- The agent exploits semantics and logs: recognizes regimes, uses priors (ranges, monotonicities), and proposes *meaningful* batches
- Loop: read logs $\Rightarrow$ summarize context $\Rightarrow$ choose sampling scheme (local sweep, bracket, or exploratory LHS) $\Rightarrow$ refine
- Outcome: better early signal, fewer wasted trials than black-box BO at similar parallelism

# Roadmap

# Concept & contributions

1. **ORFS-agent**: LLM autotuner inside ORFS; parallel trials; metric-aware refinement
2. **NL objectives**: single/multi-objective (`WL`, `ECP`); NL constraints
3. **Tool-using loop**: INSPECT / MODEL / AGGLOMERATE; optional BO for sample-efficiency
4. **Model-agnostic**: benefits scale with foundation model strength

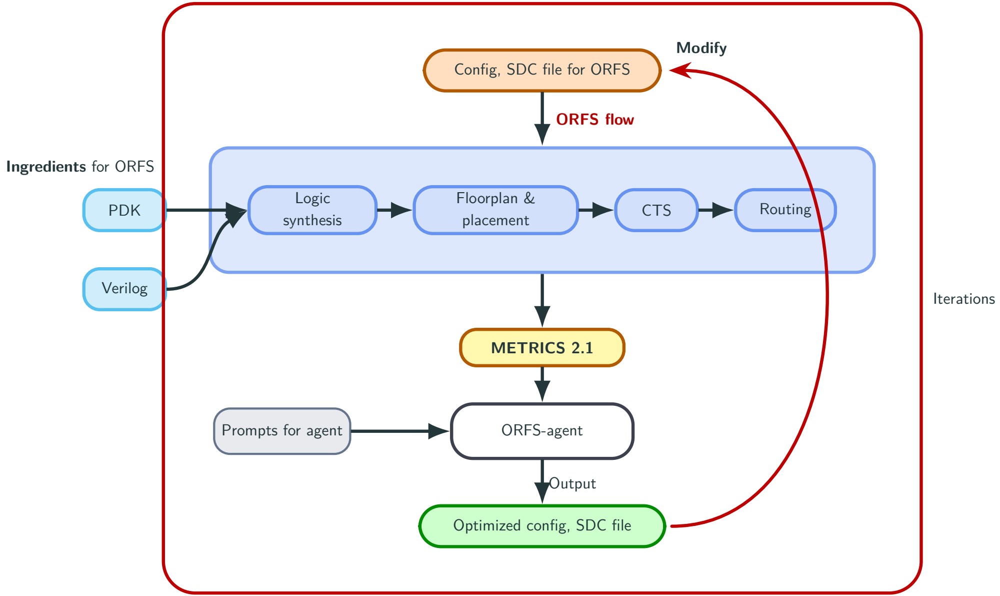
## Related work (very brief)

- **BO/autotuning**: Ray Tune, HyperOpt, Spearmint; EDA-specific OR-AutoTuner for ORFS
- **LLM + optimization**: function-calling agents and "LLM-augmented BO" (e.g., LLAMBO-style) show viability on structured domains
- **This work**: seated *inside* ORFS; uses *tool-inspected* context instead of raw CSV; supports NL objectives/constraints and parallel batches

# Roadmap

1. Motivation
2. Concept & Contributions
3. **Method**
4. Experiments
5. Results
6. Limitations & Next
7. Takeaways

# Method — ORFS-agent pipeline



# Method — objectives, surrogates, constraints

Actual metrics: $\mathrm{WL}, \mathrm{ECP}$; baselines: $\alpha$; surrogates from CTS: $'$

- Single-objective (WL): min $\dfrac{\mathrm{WL}}{\mathrm{WL}_\alpha}$

# Method — objectives, surrogates, constraints

Actual metrics: $\mathrm{WL}, \mathrm{ECP}$; baselines: $\alpha$; surrogates from CTS: $'$

- Single-objective (WL): min $\dfrac{\mathrm{WL}}{\mathrm{WL}_\alpha}$
- If WL missing: min $\dfrac{\mathrm{WL}'}{\mathrm{WL'}_\alpha}$

# Method — objectives, surrogates, constraints

Actual metrics: $\mathrm{WL}, \mathrm{ECP}$; baselines: $\alpha$; surrogates from CTS: $'$

- Single-objective (WL): $\min \dfrac{\mathrm{WL}}{\mathrm{WL}_\alpha}$
- If WL missing: $\min \dfrac{\mathrm{WL}'}{\mathrm{WL}'_\alpha}$
- Multi-objective: $\min \left(\dfrac{\mathrm{WL}}{\mathrm{WL}_\alpha} + \dfrac{\mathrm{ECP}}{\mathrm{ECP}_\alpha}\right)$

# Method — objectives, surrogates, constraints

Actual metrics: $\mathrm{WL}, \mathrm{ECP}$; baselines: $\alpha$; surrogates from CTS: $'$

- Single-objective (WL): min $\dfrac{\mathrm{WL}}{\mathrm{WL}_\alpha}$
- If WL missing: min $\dfrac{\mathrm{WL}'}{\mathrm{WL}'_\alpha}$
- Multi-objective: min $\left(\dfrac{\mathrm{WL}}{\mathrm{WL}_\alpha} + \dfrac{\mathrm{ECP}}{\mathrm{ECP}_\alpha}\right)$
- Constraints (NL): "Improve X; other metrics $\leq 2\%$ worsen"

# Method — tunables (4-var & 12-var)

- 4-var: `Core Util`, `TNS End %`, `LB Add On Place Density`, `Clock Period`
- $+8$ vars: `GP/DP padding`, `DPO enable`, `Pin/Above layer adjust`, `Flatten`, `CTS cluster size`, `CTS cluster diameter`

Conforms to METRICS 2.1 and OR-AutoTuner practice.

| Var | Range |
|---|---|
| Core Util | 20–99 |
| TNS End % | 0–100 |
| LB Add On | 0.00–0.99 |
| Clock Period | $> 0$ (ns/ps) |
| CTS size | 10–40 |
| CTS diam. | 80–120 |

# Method — tools

- **INSPECT**: PCA; min/max; correlations; plots $\Rightarrow$ context summaries (not raw CSV)

# Method — tools

- **INSPECT**: PCA; min/max; correlations; plots $\Rightarrow$ context summaries (not raw CSV)
- **MODEL**: fast fits (linear, Gaussian, GMM, etc.); LLM tunes model hyperparams

# Method — tools

- **INSPECT**: PCA; min/max; correlations; plots $\Rightarrow$ context summaries (not raw CSV)
- **MODEL**: fast fits (linear, Gaussian, GMM, etc.); LLM tunes model hyperparams
- **AGGLOMERATE**: candidate pruning via coverage/diversity; DPP-like tools; cap at 25 parallel

# Method — INSPECT: what goes into context

- **Signal shaping**: PCA/top-$k$ loadings; min/max; heavy-tail checks; outlier flags
- **Structure**: pairwise correlations (summarized, not raw matrices); simple partials where available
- **Health**: run diagnostics (timeouts, DRCs, CTS warnings) $\Rightarrow$ binary/ordinal features
- **Text $\rightarrow$ features**: log snippets $\Rightarrow$ short normalized summaries (bounded tokens)
- **Design priors**: known feasible ranges/monotonicities injected as compact hints

Aim: *succinct* context—human-data-scientist style notes, not CSV dumps.

# Method — MODEL: lightweight surrogates

- **Fits**: linear/ridge/lasso; isotonic trends; Gaussian/GMM; KDE; GP-lite on small active sets
- **Tuning**: the agent chooses hyperparameters and feature subsets; rejects unstable fits
- **Use**: rank candidates; detect diminishing returns; propose bracketing around promising modes
- **Cost ceiling**: keep per-iter modeling $\ll$ one ORFS batch; no fine-tuning of LLM itself

Surrogates guide *selection* - but they do not replace physical evaluation.

# Method — AGGLOMERATE: pick 25 diverse trials

- **Candidate pool**: $\geq 100$ feasible suggestions from heuristics + surrogates
- **Down-selection**: coverage/diversity (DPP-like scoring or $k$-medoids) under batch budget (25; 12 for JPEG)
- **Guards**: include safe baselines; cap step sizes; ensure boundary probes for space learning
- **Outcome**: low redundancy, good space coverage, and at least one conservative configuration

# Method — engineering notes (agent runtime)

- **Function-calling**: schema-checked tools; strict I/O contracts; deterministic fallbacks
- **Isolation**: containerized ORFS; pinned commits; reproducible metrics extraction
- **Parallelism**: batch launcher (25/iter); streaming log taps for early failure detect
- **Stability**: seeded sampling; retry budget; clamp out-of-range proposals
- **No model fine-tune**: benefits scale with stronger LLMs without re-training

# Roadmap

# Experiments — setup

- Nodes: **SKY130HD**, **ASAP7**; Circuits: **IBEX**, **AES**, **JPEG**
- ORFS env: pinned commit; containerized; reproducible
- Trials/iter: 25 (12 for JPEG)
- Compare: ORFS-agent (4/12 vars; with/without tools) vs OR-AutoTuner
- Resources: GCP VM (112 vCPUs, 220 GB RAM)
- Metrics: `WL`, `ECP`; also area, count, power, PDP

Open-source: github.com/ABKGroup/ORFS-Agent; Cost: $<$ \$50 at run time; $<$ \$10 now

# Reproducibility checklist

- **Code**: pinned ORFS commit; agent + tools in repo; one-command launcher
- **Env**: Docker image + exact package locks; CPU/RAM spec disclosed
- **Seeds**: fixed RNG seeds for sampling; recorded per-iter configs/metrics
- **Metrics**: `WL`, `ECP` defined/normalized vs baseline $\alpha$; surrogates documented
- **Baselines**: OR-AutoTuner config + iteration budgets published
- **Artifacts**: logs, CSVs, and JSON summaries for each iter; script to regenerate figures

# Roadmap

# Results — key outcomes

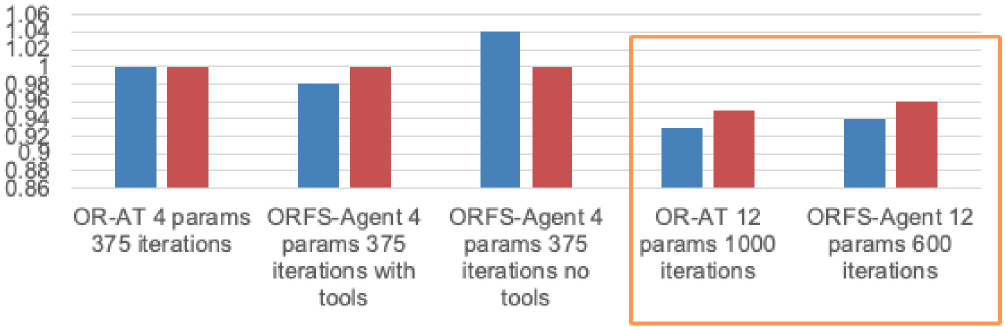



**Figure 1:** Comparison of ORFS-agent and OR-AutoTuner, with wirelength and ECP, normalizing OR-AT 4 params and 375 iterations to 1.0

Baseline: OR-AT (4 vars, 375 iters) $\equiv 1.0$. ORFS-agent: $\approx 40\%$ fewer iters; $\approx 13\%$ gains in `WL` or `ECP` (single-objective).

# Results — constrained optimization (NL prompts)

- Prompt: "Minimize `ECP`; Area/Count/Power/PDP $\leq$ 2% worse"
- Effect: primary improves; secondaries hold or improve modestly

Primary metric $\xrightarrow{\text{optimize}}$ Secondary metrics ($\leq$ 2% worsen)

Observed: $\approx$ 11% `WL` gains; $\approx$ 8% `ECP` gains (scenario-dependent).

## Prompt patterns (ready to reuse)

- **Single-objective**: *"Improve `WL`. If `WL` is unavailable, optimize the CTS surrogate* $\mathrm{WL}'$*."*

# Prompt patterns (ready to reuse)

- **Single-objective**: *"Improve `WL`. If `WL` is unavailable, optimize the CTS surrogate* $\mathrm{WL}'$*."*
- **Multi-objective (balanced)**: *"Minimize* $\frac{\mathrm{WL}}{\mathrm{WL}_\alpha} + \frac{\mathrm{ECP}}{\mathrm{ECP}_\alpha}$*. Prefer configurations that keep both terms* $\leq 1.0$*."*

# Prompt patterns (ready to reuse)

- **Single-objective**: *"Improve `WL`. If `WL` is unavailable, optimize the CTS surrogate* $\mathrm{WL}'$*."*
- **Multi-objective (balanced)**: *"Minimize* $\frac{\mathrm{WL}}{\mathrm{WL}_\alpha} + \frac{\mathrm{ECP}}{\mathrm{ECP}_\alpha}$*. Prefer configurations that keep both terms* $\leq 1.0$*."*
- **Constrained**: *"Minimize `ECP`; ensure Area/Count/Power/PDP worsen by* $\leq 2\%$*. If infeasible, return the least-violating candidate and explain."*

# Prompt patterns (ready to reuse)

- **Single-objective**: *"Improve `WL`. If `WL` is unavailable, optimize the CTS surrogate* $\mathrm{WL}'$*."*
- **Multi-objective (balanced)**: *"Minimize* $\frac{\mathrm{WL}}{\mathrm{WL}_\alpha} + \frac{\mathrm{ECP}}{\mathrm{ECP}_\alpha}$*. Prefer configurations that keep both terms* $\leq 1.0$*."*
- **Constrained**: *"Minimize `ECP`; ensure Area/Count/Power/PDP worsen by* $\leq 2\%$*. If infeasible, return the least-violating candidate and explain."*
- **Safety rails**: *"Never exceed documented ranges. If a proposal hits a boundary, include a conservative neighbor."*

# Roadmap

# Limitations & Next

- Surrogates: imperfect when detailed routing timeouts
- Heuristics: kernel/AF in GP set manually
- Next: literature integration; code-diff models in-flow; broader constraints

# What changed in last six months

- LLMs: larger contexts (to ∼1M tokens)
- Scale: ∼10k iterations feasible
- Cadence: rapid model release
- Implication: avoid brittle fine-tunes; prefer modular, model-agnostic agents

# Threats to validity & failure modes (with mitigations)

- **Routing timeouts** distort surrogates ⇒ use timeout flags; fall back to robust summaries; penalize incomplete runs
- **Nondeterminism** across stages ⇒ fix seeds; replicate key points; report variance bars
- **Metric drift** (tool/version) ⇒ pinned toolchain; normalize to concurrent baselines, not historical
- **Over-exploitation** near local modes ⇒ enforce exploratory quota in AGGLOMERATE
- **PDK/testcase leakage** of priors ⇒ keep priors coarse (ranges/monotonicities); avoid hardcoding recipe lore

# Roadmap

# Takeaways

- **Model-agnostic agent**: automates ORFS tuning; future-proof to better LLMs

# Takeaways

- **Model-agnostic agent**: automates ORFS tuning; future-proof to better LLMs
- **Competitive/better QoR**: beats BO-only, fewer iterations; co-optimization; NL constraints

# Takeaways

- **Model-agnostic agent**: automates ORFS tuning; future-proof to better LLMs
- **Competitive/better QoR**: beats BO-only, fewer iterations; co-optimization; NL constraints
- **Open & cheap**: reproducible; research-scale cost

## Takeaways

- **Model-agnostic agent**: automates ORFS tuning; future-proof to better LLMs
- **Competitive/better QoR**: beats BO-only, fewer iterations; co-optimization; NL constraints
- **Open & cheap**: reproducible; research-scale cost

QR Code linking to Drive folder with ORFS-agent and a lot more!