Offline Simulation Online Learning (OSOL)

Learn fast from cheap proxy models, then correct online using expensive trustworthy simulations

Wei Dai

March 19, 2026

Outline

1. Grid Motivation
2. Problem and Baselines
3. OSOL Idea
4. Model and Workflow
5. Intuition and Extension
6. Empirical Evidence
7. Conclusion

Fast Decisions After Grid Disturbances

- After a disturbance, a grid operator must choose a corrective action under severe time pressure.
- Candidate actions may include redispatch, reserve activation, topology switching, or inverter setpoint adjustment.
- The most trustworthy evaluation is a high-fidelity simulation of the current operating state.
- Exhaustively running that trusted simulator on every action is usually too slow for real-time use.

Two Information Sources, One Tension

- High-fidelity evidence is trustworthy but expensive, so it arrives only sequentially.
- Low-fidelity proxies are cheap and available immediately for all candidate actions.
- Proxy quality can degrade when topology, load, renewable output, or control settings change.
- We need a method that exploits proxy structure early without becoming trapped by it.

Main Question

How can we use cheap proxy information immediately, while still adapting toward the truth as expensive evidence arrives?

Ranking and Selection Under a Fixed Budget

- We view the candidate actions as a finite set of alternatives in a ranking-and-selection problem.
- The objective is to identify the best alternative using a limited budget of expensive high-fidelity evaluations.
- In the grid setting, "best" may mean the largest security margin, fastest recovery, lowest overload risk, or lowest cost subject to reliability.
- Low-fidelity data can be reduced-order simulations, screening scores, or surrogate model predictions.
- Even imperfect proxies are useful because they often preserve broad structure across alternatives.

Why Existing Baselines Are Not Enough

Full Algorithm

Re-fit the joint model whenever new high-fidelity data arrive. This is statistically flexible, but its repeated batch reprocessing becomes increasingly expensive.

Reduced Algorithm

Cluster alternatives once using low-fidelity information, then keep that structure fixed. This is efficient, but it cannot correct early proxy mistakes.

The gap between these two baselines motivates an intermediate design that is both adaptive and incremental.

OSOL: Offline Simulation Online Learning

Central Message

OSOL is not "low-fidelity instead of high-fidelity." It is "low-fidelity first, then high-fidelity correction later."

- The offline phase uses cheap proxy information to organize alternatives before expensive sampling begins.
- The online phase updates posterior beliefs and model parameters as trustworthy observations arrive sequentially.
- Incremental updates avoid restarting the estimation problem after each new sample.
- OSOL aims to retain the speed of Reduced methods and part of the adaptivity of Full re-fitting.

Mapping OSOL Back to the Grid Example

- Each alternative corresponds to one candidate operating action under the current system state.
- The trusted simulator provides expensive, state-specific evidence about the action's true quality.
- The proxy model provides an immediate reduced-order or learned score for all actions.
- Hidden structure lets related actions borrow strength even when only a few have been sampled.
- When the current context disagrees with the proxy, sequential high-fidelity evidence progressively overwrites that prior organization.

Minimal Statistical Objects

Keep Only Five Objects in Mind

- μ_i: the true but unknown quality of alternative i.
- $\tilde{\mu}_i$: the low-fidelity proxy information for alternative i.
- X_{ij}: sequential high-fidelity observations collected for alternative i.
- z_i: a latent group assignment that allows related alternatives to share information.
- $\hat{\mu}_i$: the current posterior estimate used for ranking and allocation.

Workflow Part I: Offline Organization

- OSOL begins with low-fidelity information that is available immediately for every alternative.
- It uses that proxy information to form soft latent groups rather than a hard partition.
- A Gaussian mixture is useful because it captures heterogeneous clusters, uncertain assignments, and cross-fidelity dependence.
- The output of this phase is an informed starting structure, not a final ranking.

Interpretation

Offline learning supplies a principled prior organization of the alternatives before expensive evidence begins to arrive.

Workflow Part II: Online Correction

- When a new high-fidelity sample arrives, OSOL updates the responsibility of the sampled alternative under the current model.
- The algorithm then refreshes running sufficient statistics instead of storing and reprocessing the full history.
- Parameter estimates are updated from those summaries using diminishing step sizes.
- Repeating this cycle gradually shifts the estimator from proxy-driven behavior to data-driven behavior.
- The computation remains incremental even as the total sample size grows.

Why OSOL Helps Over Time

Posterior Mean Intuition

OSOL combines three signals: the direct sample mean, shared cluster-level information, and a correction informed by low-fidelity correlation.

- Early stage: proxy structure stabilizes estimation when high-fidelity evidence is scarce.
- Middle stage: online reassignment can correct grouping mistakes that the Reduced algorithm would keep forever.
- Late stage: accumulated trusted evidence dominates, so the estimate becomes increasingly data-driven.
- Relative to repeated Full re-fitting, OSOL preserves adaptation without repeatedly solving the entire batch problem.

Theory Takeaway and ϵ-Mixing

Theory Takeaway

OSOL behaves like an online EM procedure with small corrections and diminishing step sizes, yielding stable long-run learning under standard stochastic-approximation conditions.

- Uniform sampling is analytically convenient, but it wastes budget on clearly inferior alternatives.
- ϵ-mixing preserves exploration through a uniform component that never disappears.
- The remaining budget can be directed toward competitive alternatives using an informed policy such as OCBA.
- This turns OSOL from a stable learner into a more efficient fixed-budget decision procedure.

Sequential Processing-Time Evidence

- In sequential settings, the Full algorithm must repeatedly reprocess a growing dataset after each update.
- Reduced and OSOL operate on compact summaries, so their cumulative processing cost grows much more slowly.
- In the reported sequential experiment, the Full algorithm approached 400 seconds by iteration 100, while OSOL remained below 10 seconds.
- The operational implication is that online correction can stay feasible while trusted evidence continues to arrive.

Fixed-Budget Sampling with ϵ-Mixing

- Under a fixed budget, uniform OSOL spends too many samples on clearly inferior alternatives.
- ϵ-mixing combines guaranteed exploration with informed exploitation through OCBA-style allocation.
- In the reported budget experiment, both fixed and adaptive mixing variants substantially improved probability of correct selection over uniform OSOL.
- The practical message is that stable online learning becomes more useful when allocation is explicitly budget-aware.

Takeaways and Applications

- OSOL uses cheap proxy structure immediately instead of waiting for many expensive simulations.
- OSOL corrects that initial structure as trustworthy evidence arrives sequentially.
- OSOL is incremental, memory-efficient, and compatible with fixed-budget allocation.
- The framework fits real-time decision settings such as contingency response in electric grids.
- Related applications include simulation screening, resource allocation, and other multi-fidelity decision problems.

Backup: Core Notation and Model Objects

μ_i	True unknown quality of alternative i under trusted evaluation.
$\tilde{\mu}_i$	Low-fidelity proxy features or scores available at the start.
X_{ij}	High-fidelity sample j collected from alternative i.
$z_{i,m}$	Soft membership weight linking alternative i to latent group m.
$\hat{\mu}_i$	Posterior mean used for ranking, selection, and sampling decisions.

- The fixed-budget decision rule is to select $\hat{i} = \arg\max_i \hat{\mu}_i$ after exhausting the budget.
- The main empirical target is probability of correct selection, not perfect estimation for every alternative.

Backup: Full Algorithm Summary

- Start from the full hierarchical Gaussian mixture model using both low-fidelity information and all accumulated high-fidelity samples.
- At each update, recompute responsibilities and conditional moments for every alternative using the full dataset.
- Re-estimate all mixture parameters through a batch EM cycle until the likelihood stabilizes.
- This yields strong statistical flexibility, but the per-iteration cost grows with the amount of data already collected.

Practical Limitation

Full reprocessing becomes increasingly difficult in sequential settings where observations arrive continuously.

Backup: Reduced Algorithm Summary

- Step 1 clusters the alternatives once using only low-fidelity information.
- Step 2 approximates high-fidelity parameters using those fixed cluster assignments and updated sample means.
- This keeps the computation cheap because the expensive low-fidelity clustering is not repeated.
- The limitation is structural rigidity: if the initial low-fidelity grouping is wrong, later high-fidelity data cannot repair it.

Key Contrast with OSOL

OSOL preserves incremental computation while still allowing the latent structure to move over time.

Backup: Proof-of-Concept Convergence

- The proof-of-concept study used 100 alternatives, 3 latent components, and observation variance $\sigma^2 = 10$.
- With step sizes proportional to $n^{-0.6}$, the parameter updates became small after roughly 75 iterations.
- Posterior prediction quality remained strong despite the noisy setting, with reported MSE 0.7971 and MAE 0.7001.
- The experiment supports the basic claim that online EM can recover useful structure without batch re-fitting.

Backup: I.I.D. Comparison with Batch Baselines

- In moderate and large-scale i.i.d. comparison settings, Full, Reduced, and OSOL all reached PCS near 1.00 after enough data were observed.
- OSOL's estimation error remained close to the batch methods, showing that online processing did not sacrifice asymptotic accuracy.
- The timing gap remained substantial: for the large-scale scenario, Full required about 5.889 seconds per iteration while OSOL required about 5.99×10^{-3} seconds.
- The main interpretation is that OSOL can approximate batch accuracy while remaining far more scalable.

Backup: Clustering Failure Scenario

- The failure scenario places the best-performing component adjacent to the worst component in low-fidelity space.
- Fixed low-fidelity clustering therefore creates a misleading prior organization for the Reduced algorithm.
- OSOL can revise responsibilities as high-fidelity evidence accumulates, which helps correct that mismatch.
- In the reported study, OSOL reached PCS around 0.88 by iteration 80, compared with about 0.84 for Reduced and about 0.45 for the high-fidelity-only baseline.

Backup: Additional ϵ-Mixing Details

- The budget-controlled study allocated 800 total high-fidelity observations across 50 designs.
- After a minimal initialization stage, the remaining budget was allocated sequentially under uniform, fixed-mixing, or adaptive-mixing policies.
- The fixed policy used $\epsilon = 0.7$, while the adaptive policy increased exploitation gradually as confidence improved.
- Both mixing strategies outperformed uniform OSOL in final probability of correct selection, illustrating the value of explicit exploration-exploitation control.