PROJECT #21: Nuclear Power Systems for AI Data Centers (Stage 1-4)

STAGE 1: UCDP v9.3-SFE VISION GENERATION

Input: Main Idea: Prototype end-to-end nuclear-based power system proxies to enable secure, sovereign energy infrastructure for hyperscale AI data centers in national security missions. Run Mode: Standard (Full Protocol) Sector: Defense/Nuclear Security

1.
Cross-Domain Fusion with Weak Signals:

◦ Fused with 3 distant domains: hydroelectric dam operations (baseload energy management), symphony orchestration (system synchronization), vineyard irrigation (adaptive resource allocation).

◦ Generated 3 breakthrough concepts with Leap-Ahead Impact Statements:

▪ Hydroelectric Baseload AI: Fuses hydroelectric dam baseload management with AI to design nuclear–AI campus architectures targeting 10x-class improvement in modeled power-delivery reliability to AI data centers (vs. current grid-only baselines) under MARVEL-style microreactor and PPA proxy scenarios, leveraging the shared structure of flow/flux differential equations used in both dam dispatch models and neutron transport/diffusion approximations.

▪ Symphony Synchronization Models: Applies orchestration harmony to AI for 10x-class efficiency in multi-reactor coordination in energy campuses, emphasizing score-level global optimization across multi-reactor nuclear-plus-data-center sites such as TMI-style campuses, rather than unit-level heuristics.

▪ Vineyard Adaptive Systems: Integrates irrigation adaptation for 10x dynamic power scaling in response to AI compute demands, mapped to DOE microreactor and MARVEL-style scenarios where modular reactors ‘irrigate’ clusters of modular data centers with adaptive power output.

2.
Top Concept Selection:

◦ Most promising: Hydroelectric Baseload AI.

◦ One-line summary with quantified impact: Fuses hydroelectric dam baseload management with AI to create nuclear power systems for AI data centers targeting 10x-class improvement in modeled reliability of power delivery, aiming for ≥99% modeled availability for critical AI workloads under simulated outage scenarios, and scenarios where ≥50–70% of power is supplied by on-site nuclear or microreactors, with the grid treated as backup.

◦ Next-Best Action (blue sky version): Launch an unclassified LANL proxy pilot within 3–6 months using public dam dispatch and reservoir datasets to train AI controllers that stabilize baseload delivery under MARVEL-style microreactor and modular data center load curves, producing metrics directly comparable to DOE microreactor and grid-interaction studies.

3.
Technical Correlation Explanation: Neutron transport equations ($\phi = \frac{1}{v} \frac{\partial \Psi}{\partial t} + \Omega \cdot \nabla \Psi + \Sigma_t \Psi = \int \Sigma_s \Psi' d\Omega' + S$) model neutron flux in nuclear reactors, and hydrological/dispatch models for dams use structurally similar conservation and flow equations; this shared structure allows AI to transfer methods for stabilizing baseload supply under variable demand. For power-reactor planning, diffusion-form approximations of the transport equation are typically used, which further clarifies the analogy to bulk water-flow and reservoir-balancing equations. This correlation is non-obvious as dams emphasize water resource management, while nuclear power handles fission reactions, yet both use differential equations for steady-state flow control. By using well-characterized nuclear PPAs and microreactor programs as boundary conditions, this project frames nuclear–AI campuses as a sovereign-compute foundation consistent with emerging hyperscaler–nuclear deals and DOE microreactor validation work.

STAGE 2: PROXY IMPLEMENTATION

This stage translates the vision into executable proxies, focusing on unclassified simulations to test baseload stabilization. Using USGS streamflow data as analogs for nuclear flux, we build AI controllers in Torch to optimize delivery under variable loads (e.g., AI training spikes). No real nuclear data is used—proxies only.

1. Proxy Setup and Data Acquisition:
    - Use the USGS Water Services / NWIS APIs to pull streamflow time-series (discharge in CFS) for selected gages; store data as `timestamp, flow_cfs, stage_ft` in CSV or Parquet for replayable experiments.
2. AI Controller Development:
    - Simple linear model (as bundle) for diffusion approx; extend to RNN/LSTM for time-series forecasting if needed.
    - Add a simple LSTM-based controller that maps recent `flow_cfs` history and a synthetic AI load profile to a dispatch decision for each time step.
    - Define a scalar cost function ($J = \sum_t \alpha \cdot \text{unserved\_MWh}_t + \beta \cdot |\Delta P_t|$) and evaluate controllers by their ability to reduce (J) versus a non-AI baseline dispatch.
3. Bundle and Metrics:
    - See Production Deployment Bundle below for core proxy code.

◦ Metrics as targets (not measured); compute runtime as proxy for efficiency.

Production Deployment Bundle **baseload_ai_deploy.py**:

```
import pandas as pd
import torch
import numpy as np
import time
import gc

def
deploy_baseload_proxy(csv_file='usgs_streamflow_timeseries.
csv', flow_column='flow_cfs'):
    data = pd.read_csv(csv_file)  # Load public USGS
streamflow proxy data
    flow = data[flow_column].values  # Proxy for
dispatchable flow
    start = time.perf_counter()
    for ep in range(100):  # 100 iterations
        model = torch.nn.Linear(1, 1)  # Simple proxy for
diffusion approx
        out = model(torch.tensor(flow,
dtype=torch.float32).unsqueeze(1))
    duration = time.perf_counter() - start
    metrics = {
        'target_reliability_increase': 0.20,  # 20% modeled
gain goal in analogs
        'target_availability': 0.99,          # ≥99% modeled
availability goal
        'iterations_100_seconds': duration
    }
    # Simple rollback flag placeholder (real logic in
analysis layer)
    metrics['rollback_triggered'] = not
np.isfinite(duration)
    print(metrics)
    # Outputs feed energy pipeline via Python API
    del model, data; gc.collect()
    metrics['gc_total'] = gc.get_stats()['total']  # Proxy
to bound compute cost
    return metrics
```

```python
if __name__ == '__main__':
    metrics = deploy_baseload_proxy()
    print(metrics)
```

4. Initial Testing:
   - Run on sample USGS data; validate against baselines (e.g., non-AI mean flow).

STAGE 3: VALIDATION & SCALING

This stage validates proxies against DOE benchmarks (e.g., MARVEL sims) and scales to multi-reactor scenarios. Focus on modeled metrics; no real hardware.

Validation Criteria (replace existing bullets): "Reliability: In simulation, BaseloadAI should achieve ≥99% availability for high-priority AI loads across at least 100 outage scenarios."

"On-site share: Demonstrate scenarios where 50–70% of power is supplied by simulated on-site nuclear units, with the grid acting as backup."

"Performance: Show approximately 20% reduction in expected unserved AI load versus a simple heuristic dispatch baseline, measured using the cost function (J)."

Modeling approach (replace tool name-drops): Use a diffusion-style neutronics model to generate feasible nuclear power trajectories under ramp and safety constraints, coupled to a simple microgrid simulator that includes AI data center loads and grid import/export limits.

Multi-reactor "symphony" (replace the PyG-specific line): Represent a campus of 2–5 reactors and multiple data centers as a graph, with nodes for reactors, data centers, and grid interconnects and edges for transfer limits, and apply a campus-level optimization step that allocates load to minimize (J).

Go/No-Go:

- At 180 days: 20% modeled gain confirmed; if not, rollback to Stage 2.

STAGE 4: DEPLOYMENT ROADMAP

This stage outlines phased rollout from proxies to classified integration, aligned with DOE timelines.

BaseloadAI is deployed as a planning and decision-support tool, not as a direct reactor control system. Operators use its dispatch policies and risk metrics to inform campus design and operating procedures, which are implemented through existing plant and grid control systems.

Phase 1 (3–6 Months): LANL proxy pilot using USGS streamflow and synthetic AI load profiles to train and evaluate BaseloadAI controllers against a non-AI baseline using the cost function (J).

Phase 2 (6–12 Months): Integrated simulations combining diffusion-style neutronics models, microgrid power-flow models, and BaseloadAI controllers to test reliability and on-site nuclear share targets across many outage and load scenarios.

Phase 3 (12–24 Months): Multi-reactor campus simulations (2–5 units) with symphony-style coordination and vineyard-style adaptive throttling to match dynamic AI training and inference loads while minimizing (J).

Phase 4 (24–48 Months): If unclassified simulation results justify it, port BaseloadAI workflows to DOE/NNSA HPC systems to run large ensembles of nuclear–AI campus scenarios and, where appropriate, evaluate recommended policies in microreactor or microgrid testbeds under strict operator and safety oversight.

This roadmap aligns with ongoing DOE/NNSA microreactor and HPC efforts without naming specific programs or regulations.

NNSA: LANL lead, Five Eyes allies and DOE Microreactor/MARVEL collab, 1.5–2.5M Y1, RFI-ready via AI–hydro proxies, PPA/microreactor scenarios, and neutronics-informed baseload models.

Click here to open the open source, AI assisted technical exploration and supplemental starter kit. *(Manual link: https://ucdpforge.com/pdfs/R2I/NNSA_Genesis_21_IG_NucAIDCPwrCampus-Stage-1-4.pdf*