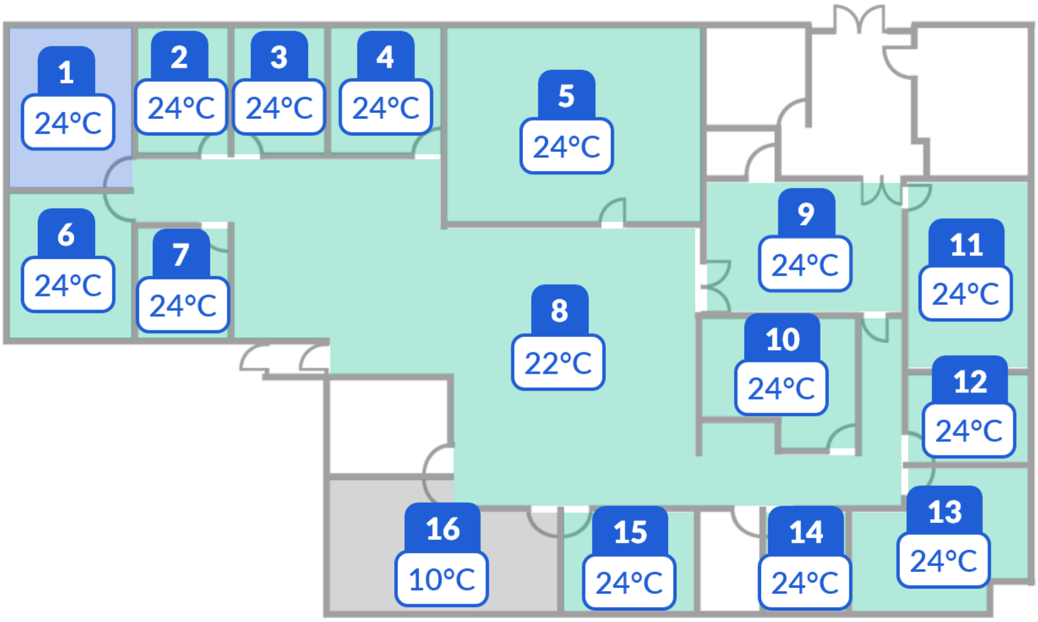
NEST: Towards an Autonomous, Learning-Based Edge Architecture for Energy-Efficient Building Control



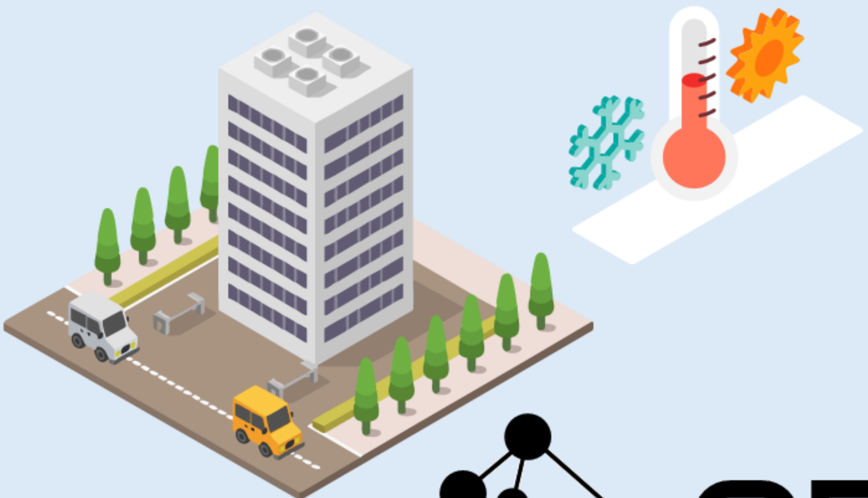


Table of Contents

Versioning

This document is subject to version control to ensure full traceability of changes. Each update is recorded with its author, date, and a short description of the modifications.

Version	Date	Author	Organization	Change Description
1.0	2025/10/19	Youssef Menjour	Graiphic	First publication of the NEST Whitepaper

Abstract

This paper introduces NEST (Next Energy Smart Twin), an edge-native control architecture designed to maximize building energy efficiency through a staged integration of generative AI, data-driven digital twins, and reinforcement learning. Unlike traditional Building Management Systems (BMS), which rely on static rules or hard-coded optimization strategies, NEST enables a building to progressively learn its own thermal behavior and optimize its energy usage without cloud dependency.

The system operates in three phases. **First**, a large generative language model (LLM) is used as a temporary agent to initiate intelligent HVAC control, using real-time environmental, occupancy, and energy data. This bootstraps the system while passively collecting high-frequency telemetry. **In the second phase**, a neural network is trained to replicate the building's thermal dynamics, effectively acting as a digital twin. **Finally**, a reinforcement learning agent is trained on this surrogate environment to generate optimal control policies that minimize energy use while maintaining comfort constraints.

NEST is fully embedded at the building level, relying on local inference hardware (e.g. DXG Spark) and on-site databases for data collection and model execution. This ensures low-latency, privacy-preserving, and scalable deployment across diverse building typologies. The proposed architecture challenges the current paradigms of cloud-based smart building systems, and points toward a future where each building becomes an autonomous energy agent, capable of continuous self-optimization.

Introduction

Buildings account for approximately **40% of global energy consumption** and are responsible for a significant portion of greenhouse gas emissions. As climate targets tighten and energy prices fluctuate, optimizing building energy performance, especially for HVAC systems has become a critical research and operational priority. Despite the proliferation of Building Management Systems (BMS), most real-world deployments remain simplistic, relying on static rules or manually configured schedules that poorly adapt to changing environmental, usage, and occupant conditions.

Recent research has explored more advanced strategies for building control, including Model Predictive Control (MPC), digital twins, reinforcement learning (RL), and more recently, generative AI agents such as large language models (LLMs). While MPC can provide precise optimization, it often requires detailed modeling and is not easily transferable between buildings. RL can learn adaptive policies but suffers from high sample complexity and cold-start limitations. LLMs offer promising flexibility and general knowledge reasoning, but their lack of determinism, safety guarantees, and long-term learning capacity make them unsuitable as sole control agents in safety-critical settings.

In this work, we present NEST, **a three-stage edge-native architecture that enables a building to progressively learn to optimize itself through data**. The system transitions through three levels of autonomy:

1. A generative AI agent is initially used to pilot the building based on environmental and occupancy data, issuing intelligent HVAC controls via prompt-based reasoning.
2. A data-driven digital twin is then trained using the data accumulated during the first phase. This model does not take control but instead simulates the building's thermal behavior and serves as the environment for training the third-stage reinforcement learning agent.
3. A reinforcement learning agent is finally trained within this surrogate model to learn an optimal control policy that balances comfort and energy minimization.

All computation and control loops are embedded at the edge (e.g., on a DGX spark), ensuring low latency, data sovereignty, and robustness without relying on cloud infrastructure. NEST enables each building to become an autonomous energy agent capable of continuous self-improvement, adapting to its own thermal dynamics and usage patterns over time.

In the following sections, we detail the system architecture, learning pipeline, control strategies, and discuss how this approach addresses limitations of current state-of-the-art methods.

Related Work

The increasing complexity of building systems and the demand for energy efficiency have led to the emergence of advanced control strategies beyond rule-based logic. Among them, Model Predictive Control (MPC), digital twins, reinforcement learning (RL), and more recently large language models (LLMs) have each brought unique capabilities and limitations.

MPC is a widely studied technique that uses an explicit physical or empirical model to predict future building states and optimize control actions over a horizon. While effective in theory, MPC suffers from high modeling costs and requires extensive calibration, making it impractical for widespread deployment across heterogeneous building types. Its transferability is limited, and its performance can degrade under model mismatch or changing operating conditions.

Digital twins have gained significant attention as virtual representations of physical buildings that integrate real-time sensor data with static models. When calibrated properly, they provide a powerful basis for simulation, anomaly detection, and predictive analytics. However, most digital twin implementations remain passive: they mirror the building state but do not directly close the control loop. Furthermore, they are often centralized, relying on cloud-based infrastructure, which introduces latency and raises privacy concerns. Many existing twins also depend heavily on detailed BIM (Building Information Modeling), which may not be available for older or smaller buildings.

Reinforcement learning has been applied to building control due to its ability to learn from experience and adapt to complex environments. RL agents can, in principle, learn optimal control policies without requiring an explicit building model. However, they are notoriously sample-inefficient, requiring hundreds or thousands of episodes to converge. This limits their viability in real buildings, where exploration can be costly or unsafe. Recent works have attempted to train RL agents in simulation first, often using digital twins as proxies, before transferring to real deployment.

The emergence of generative AI and LLMs introduces a new paradigm for control: prompt-based reasoning. Pre-trained models such as GPT-3.5 or GPT-4 have been shown to make plausible HVAC control decisions with minimal fine-tuning. In simulation environments like EnergyPlus, LLMs have achieved energy savings in the same range as trained RL agents, despite having no prior exposure to the building in question. Their ability to reason with general knowledge makes them attractive for zero-shot or low-data settings. However, they lack consistency, cannot learn from new data unless explicitly re-prompted, and are not deterministic, making them unsuitable for long-term or safety-critical control. Most importantly, they require cloud APIs, which are not feasible for embedded, low-latency deployments.

In this context, NEST builds on the strengths of these paradigms while addressing their limitations. It uses a generative agent only as a temporary bootstrapping layer, transitions toward a local, continuously trained digital twin, and ultimately adopts a reinforcement learning policy that is both customized and embedded. Unlike previous approaches, NEST operates entirely on the edge, integrates learning directly into the control loop, and ensures long-term adaptability and autonomy.

System Overview

NEST is designed as a fully autonomous energy management architecture that enables buildings to progressively learn and optimize their own operation over time. Its core principle is a staged, hybrid intelligence pipeline combining generative control, supervised learning, and reinforcement learning, all executed locally on embedded hardware. This section presents the high-level system design, the data flow across learning phases, and the technical components that support autonomy, privacy, and robustness.

General Workflow

The NEST pipeline is structured in three stages, each building on the previous:

1. **Generative AI Bootstrapping**: At deployment, a pre-prompted large language model (LLM) serves as an initial control agent. It reads the building's current internal temperature, outdoor conditions, scheduled occupancy, and user-defined comfort ranges. Based on this context, it generates HVAC setpoints to reduce energy use while respecting comfort tolerances. This stage allows the building to begin intelligent operation without prior training or system identification.

2. **Digital Twin Learning:** While the generative agent is active, a parallel process collects high-resolution sensor data every three minutes. After an initial data accumulation period (for example, one month), a supervised learning model is trained to replicate the building's thermal response. This model constitutes the first version of a predictive digital twin capable of simulating the effects of control actions under varying environmental and occupancy conditions.

   Importantly, data collection continues continuously. The digital twin is retrained at regular intervals, typically once per month, by replacing the previous model with a new one trained on the enlarged dataset. As the amount and variety of data grow over time, the digital twin becomes increasingly accurate and better aligned with the building's real thermal behavior.

   Because all collected data is timestamped, the system naturally captures seasonal effects such as solar gain, wind orientation, external temperature trends, and occupancy variations over the year. This accumulation allows the model to learn not just short-term dynamics, but long-term temporal patterns across different periods and operating modes.

   This iterative learning process ensures that the control system stays responsive and adapted to evolving patterns. All data and models are managed locally on the edge device, preserving privacy and ensuring autonomous operation without any dependency on cloud infrastructure.

3. **Reinforcement Learning Optimization**: Once the digital twin is operational, a reinforcement learning (RL) agent is trained entirely in simulation to discover optimal HVAC control policies. The agent maximizes a custom reward function that penalizes energy consumption while encouraging compliance with comfort constraints.

   As the digital twin is periodically retrained with newly collected data, the RL agent is also updated accordingly. Each new version of the twin provides a more accurate and realistic simulation environment, allowing the RL agent to refine its policy and improve its performance over time. This iterative process enables the control strategy to stay aligned with the evolving behavior of the real building.

   When training reaches a satisfactory level of convergence, the RL agent replaces the generative controller and takes over real-time building control using local policy inference. This allows the building to operate autonomously with a continuously improving control policy tailored to its specific thermal and usage characteristics.

Embedded Infrastructure

All stages of the NEST pipeline run entirely on a local embedded computing node, now based on the NVIDIA DGX Spark platform released in October 2025. This compact and powerful AI device includes 128 GB of unified memory, up to 4 TB of NVMe SSD storage, and dedicated hardware acceleration for training and inference using ONNX Runtime.

The device is installed within the building and performs multiple integrated functions. It connects to internal environmental sensors that capture zone-level air temperature, with optional measurements of humidity, carbon dioxide concentration, and occupancy levels. It also integrates external environmental data from weather APIs or dedicated sensors that measure outdoor temperature, wind conditions, and solar radiation, including direct irradiance or cloudiness levels.

The system actively controls the HVAC installation by adjusting setpoints, selecting operational modes such as heating, cooling, or ventilation, and modulating the fan speed or output power of air delivery systems.

All collected data is timestamped and saved locally in a time-series database. This includes internal and external environmental conditions, comfort parameters, and the energy consumed by HVAC equipment at each sampling point. The time references make it possible to capture daily cycles, work hours, weekends, holidays, and seasonal effects such as heating demand in winter or solar load in summer.

The DGX Spark executes the full intelligence pipeline locally, including generative model inference, training and serving of the supervised digital twin, and training and deployment of the reinforcement learning agent. All AI components are managed using the Graiphic framework, which is optimized for ONNX compatibility and efficient edge deployment.

A secure web interface hosted on the same device provides monitoring, diagnostics, historical data visualization, and manual override capabilities if needed. Because all operations are handled locally, the system offers low latency, preserves data privacy, and enables scalable deployment across multiple buildings without requiring cloud infrastructure.

Interoperability and Modularity

The system is protocol agnostic and can be integrated with existing building automation systems using standard interfaces such as Modbus BACnet MQTT or REST APIs. This makes it compatible with most buildings regardless of their existing infrastructure and avoids the need to replace legacy control systems.

Each intelligence layer in the NEST pipeline including the generative agent, the digital twin and the reinforcement learning policy is encapsulated as an independent control module. These modules communicate through a unified interface that is configured during deployment using a context aware prompt. This initial prompt defines the building specific elements such as the available sensors, the control capabilities, the occupancy patterns and the comfort targets.

The internal pipeline including data collection model training and control execution is automatically adapted to each building. The time series database reflects only the variables that are actually available in the building, ensuring a minimal and relevant footprint. The training and inference workflows are also customized to match the building's behavior thermal response and actuation logic. As a result, the solution is tailored to the real characteristics of each site without requiring manual reprogramming or cloud-based configuration.

This modular and flexible design allows for future upgrades including the integration of alternative control strategies such as model predictive control or the addition of new optimization layers. All modifications can be made without reengineering the entire system.

In summary NEST provides a self-contained adaptive and non-intrusive control framework that can be deployed quickly in a wide range of buildings. It leverages existing equipment, supports continuous learning and delivers intelligent energy management without disrupting current operations.

Phase 1: Generative AI-Driven Bootstrapping

The first phase of NEST addresses a critical challenge in autonomous building control: how to begin intelligent operation when no historical data or calibrated model is yet available. Instead of relying on predefined rules or waiting for a digital twin to be built, NEST leverages a generative AI agent, specifically a pre-trained large language model (LLM) to initiate real-time HVAC control based on semantic reasoning.

Role of the Generative Agent

The LLM is used as an initial supervisory controller capable of interpreting the building’s current context and suggesting reasonable control actions. Prompted with real-time data, including zone temperatures, external weather conditions, scheduled occupancy, time of day, and user-defined comfort bands, the model outputs HVAC setpoints (temperature targets and ventilation intensity). The prompts are engineered to simulate the reasoning process of an expert building manager, encouraging the LLM to prioritize energy efficiency while maintaining thermal comfort.

This strategy enables the system to take meaningful control actions from day one, even in the absence of a trained model. Crucially, it also enables the collection of paired (state, action, outcome) data points necessary for supervised learning in the next phase.

Sensor and Input Data

The NEST system relies on a combination of internal and external data sources, all timestamped to capture both short-term variations and long-term seasonal effects. Internally, the building is equipped with temperature sensors deployed across thermal zones. These sensors provide real-time air temperature measurements that guide HVAC control. The system also controls HVAC actuators directly, including the temperature setpoint per zone, the operating mode selection (heating, cooling, or ventilation), and the fan power level.

Externally, NEST receives environmental data either from meteorological APIs or from local outdoor sensors. These inputs include outdoor air temperature, wind speed and direction, cloud cover conditions, and solar radiation levels. In addition to external weather feeds, optional on-site radiation sensors can provide direct measurements of irradiance for more accurate modeling of envelope heat transfer via radiation, conduction, and convection.

Each data point is timestamped to preserve contextual information such as time of day, day of the week, and seasonal cycle. This temporal structure allows the system to capture effects related to occupancy schedules, business hours, weekends, and holidays, and to account for variations between winter and summer operation. All collected data is stored locally in a time-series database and used to feed both the predictive digital twin and the learning-based control components of the NEST architecture.

Prompt Engineering and Output Formatting

To interface with the LLM, sensor data is converted into structured natural language prompts. For example:

"At 10:15 AM, Zone 1 has an internal temperature of 22.1°C, with an outdoor temperature of 18.3°C and forecast rising. The room will be occupied until 6 PM with a comfort band of 21.5–23.5°C. Power draw has increased by 6% compared to the previous cycle. Suggest an optimal temperature setpoint and ventilation level."

The model's response is parsed to extract actionable parameters and enforce domain-specific constraints (e.g., allowable setpoint ranges). Determinism is ensured by setting the generation temperature to zero and constraining outputs to predefined formats (e.g., JSON or key-value responses).

Benefits and Limitations

Using a generative agent offers several advantages:

- **Cold-start bypass**: immediate deployment with no prior training or modeling.
- **Contextual reasoning**: the LLM incorporates commonsense logic and HVAC heuristics.
- **Data generation**: every decision becomes a supervised data point for future training.

However, limitations exist:

- LLMs cannot learn incrementally from data without retraining.
- They may produce inconsistent or suboptimal actions if prompts are ambiguous.
- Their reliance on textual inference makes them ill-suited for long-term closed-loop control.

For these reasons, the generative agent is used as a temporary bridge to the second stage of NEST: the construction of a predictive digital twin.

Phase 2: Data-Driven Digital Twin Construction

The second phase of NEST focuses on building a predictive digital twin of the building's thermal behavior using real data collected during the generative control phase. This model does not replace the generative agent in control but instead serves as a simulation environment to safely train a reinforcement learning agent in the next phase. This model referred to as the digital twin captures the internal thermal dynamics of the building, enabling simulation, forecasting, and ultimately, the safe training of control policies. Unlike BIM-based or physics-first digital twins, the NEST twin is purely data-driven and trained from real measurements collected during the bootstrapping phase.

Objective and Model Structure

The goal of the digital twin is to learn the building's response to control actions under varying conditions. The model takes as input the current internal temperature, HVAC setpoints, external weather data, time, occupancy schedules, and recent history, and outputs the predicted temperature evolution and energy consumption for the next time step. Formally, the digital twin approximates a function:

T_{t+1}, P_{t+1} = f(T_t, T_{t−1}, W_t, S_t, O_t)

where T is internal temperature, W is weather, S is the HVAC setpoint vector, and O encodes occupancy and schedule information. P is power consumption.

Depending on the data richness and temporal granularity, the model may use either a feed-forward MLP, an LSTM or GRU-based recurrent network, or a 1D convolutional architecture for time series encoding.

Training Process

After approximately one month of operation, the system has amassed tens of thousands of timestamped data points (typically one per zone every 3 minutes). The training set is constructed by aligning each control decision with its observed temperature and power outcome in the subsequent time step. The training pipeline includes:

- data cleaning and outlier removal (e.g., from faulty sensors),
- normalization and feature engineering (e.g., time-of-day embeddings, lagged variables),
- supervised regression training with MSE or Huber loss,
- validation on hold-out days (e.g., weekends or holidays with different profiles).

The model is deployed locally on the same edge device using ONNX or TensorRT for efficient inference.

Adaptive Retraining

To prevent model drift and accommodate seasonal shifts, the digital twin is periodically retrained using an expanding or sliding window of recent data. This ensures that the model remains synchronized with actual building behavior, including changes in occupancy, equipment degradation, or weather patterns. Retraining can occur offline during low-activity hours or opportunistically in the background.

Role Within the System

The trained twin becomes the simulation environment for Phase 3, allowing reinforcement learning agents to experiment freely without affecting real occupants or equipment. Additionally, it can be used in the operational phase to:

- predict near-term temperature evolution under candidate setpoints,
- estimate energy consequences of user preferences,
- support fault detection by comparing real and predicted behavior,
- or generate synthetic data for pre-training future controllers.

Comparison to Conventional Twins

Unlike conventional digital twins that rely on physics-based modeling (e.g., EnergyPlus, TRNSYS), NEST's twin requires no architectural documentation or calibration effort. It is entirely driven by observed data, which makes it more scalable and adaptive, albeit at the cost of lower physical interpretability. However, the model's predictive performance can be validated continuously against reality, ensuring it remains a faithful proxy of the building.

This data-driven digital twin is the linchpin of NEST's learning loop, enabling the safe and efficient development of customized control policies tailored to the unique dynamics of each building.

Phase 3: Reinforcement Learning for Optimal Control

In the final phase of the NEST pipeline, the system transitions from predictive simulation to autonomous optimization. The trained digital twin now serves as a reliable environment in which a reinforcement learning (RL) agent can explore and learn optimal control strategies. The RL agent replaces both the generative AI controller and static setpoint schedules, offering a fully adaptive policy tailored to the building's unique behavior.

Problem Formulation

The building control problem is formulated as a Markov Decision Process (MDP), where:

- **State (s_t)** includes internal temperatures, external weather conditions, forecast variables, occupancy schedules, and recent control history;
- **Action (a_t)** is a vector of HVAC setpoints and ventilation intensities to apply;
- **Reward (r_t)** combines energy efficiency and thermal comfort.

A typical reward function is:

r_t = −α · P_t − β · ∑_z |T_{z,t} − T_{z,target}|

where P_t is the total power consumed, and the second term penalizes deviation from comfort targets across all zones z. Weights α and β reflect the trade-off between efficiency and comfort.

Agent Design and Training

The RL agent is trained using the digital twin as a simulator. Depending on the action space and building complexity, the algorithm may use:

- **DQN** (for discrete setpoints),
- **PPO or SAC** (for continuous action spaces),
- or hybrid approaches with safety constraints (e.g., safe RL).

Training is conducted entirely offline and locally, using multiple rollouts of simulated building operation. The twin allows the agent to test thousands of action sequences without impacting real equipment or comfort.

Key design components include:

- exploration strategy with epsilon decay or entropy regularization,
- policy regularization to avoid overreactive control,
- curriculum training starting from baseline schedules.

Deployment and Inference

Once trained, the RL policy is frozen and deployed to the local controller. Every 3 minutes (or user-defined interval), the agent reads the current state from sensors and digital context, then outputs the next optimal action. Inference latency is minimal (<50 ms on embedded hardware). A post-processing layer ensures safety constraints (e.g., HVAC rate limits, ventilation thresholds) are respected before issuing commands to the actuators.

Continual Learning and Re-Training

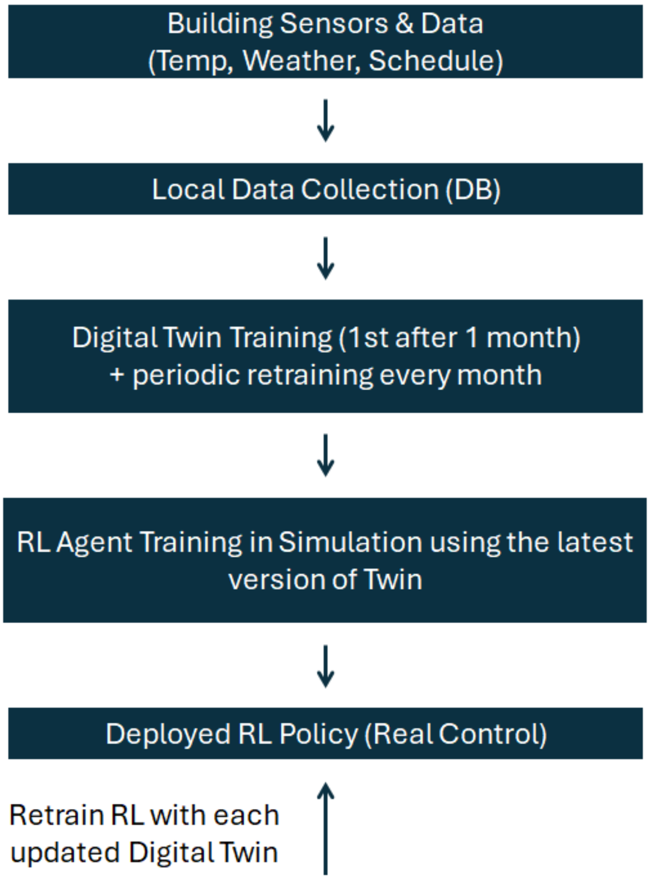


Figure 1 – illustrates the full NEST learning pipeline, showing how the digital twin and RL policy are continuously updated as more data accumulates. This closed-loop process enables autonomous, adaptive control that improves over time.

To maintain relevance as the building evolves, the RL agent can be periodically retrained using new rollouts from the updated twin. Alternatively, multiple policies can be maintained and switched based on season or occupancy mode (e.g., "winter day", "weekend standby").

If model drift is detected—e.g., predicted temperatures deviate from observed values beyond a tolerance—the system can automatically fallback to the generative controller or a conservative baseline policy while retraining is underway.

Advantages Over Static and Heuristic Controllers

Compared to fixed rule-based systems or initial LLM control:

- the RL agent **minimizes energy consumption** with higher precision,
- **adapts to complex non-linear dynamics**, including thermal lag and solar gains,
- **learns building-specific strategies** that generalize poorly across sites (e.g., pre-heating before arrival, coordinated zone setbacks),

- and enables **closed-loop learning** in a fully autonomous manner.

While reinforcement learning has historically been challenging to deploy in real buildings due to sample inefficiency and safety risks, NEST's use of a data-driven surrogate environment resolves both, allowing safe, accelerated policy development tailored to each site.

Edge-Centric Architecture

A central design principle of NEST is that all sensing, learning, and control operations are performed locally within the building. This edge-centric approach ensures autonomy, data privacy, real-time responsiveness, and scalability across large building portfolios without relying on external infrastructure.

Hardware Platform

Each building is equipped with a compact embedded computing unit, such as the NVIDIA Jetson Nano, Jetson Orin, or the high-performance DGX Spark platform released in October 2025. This on-site hardware includes up to 128 GB of unified memory, several terabytes of local SSD storage, and built-in acceleration for AI inference and model training using ONNX Runtime.

The system continuously collects data from internal sensors, including zone-level air temperature, humidity, and occupancy. It also integrates external environmental inputs from weather APIs or dedicated outdoor sensors, which provide real-time information on outside temperature, wind conditions, and solar radiation, including irradiance and cloud coverage levels.

NEST controls the building's HVAC system by adjusting temperature setpoints, selecting operational modes such as heating, cooling, or ventilation, and modulating fan power. All control commands and sensor measurements are timestamped and stored in a local time-series database. This allows the system to reconstruct thermal and usage history over time, including variations linked to time of day, occupancy schedules, weekends, holidays, and seasonal conditions.

All components of the NEST intelligence pipeline operate on the same embedded platform. This includes the generative controller, the digital twin, and the reinforcement learning model, all managed through the Graiphic framework, which is optimized for ONNX-based execution on edge hardware.

A secure web interface is available for facility managers to monitor real-time data, inspect historical trends, review control decisions, and apply manual overrides if needed. This local-first architecture ensures low-latency response, preserves data privacy, and allows for scalable deployment across a variety of building configurations without relying on external cloud services.

Interfacing and Control

NEST interfaces with the building's HVAC system and BMS through standard protocols such as:

- Modbus RTU/TCP,
- BACnet/IP,
- MQTT,
- or vendor-specific APIs (for smart thermostats, ventilation units, etc.).

A low-level driver layer abstracts these communication protocols into a unified interface, allowing the control modules (LLM, twin, RL agent) to issue setpoints in a hardware-agnostic fashion.

The same layer also applies safety checks and command rate limits to avoid aggressive switching or equipment wear.

Real-Time Loop and Latency

All control decisions are executed on a fixed cycle (typically every 180 seconds). The system's latency budget is optimized as follows:

- data acquisition and pre-processing: <500 ms,
- model inference (RL or LLM): <50 ms,
- control validation and dispatch: <100 ms.

This ensures that NEST can respond to environmental changes in near real time while respecting the dynamics of HVAC equipment (e.g., ramp rates, cycle durations).

Monitoring and Override

A lightweight local web server (e.g., Flask, Node.js) is embedded on the edge device. It allows authorized users (facility managers, technicians) to:

- visualize real-time sensor values and control actions,
- inspect historical performance and energy usage,
- trigger manual overrides or fallbacks (e.g., revert to safe schedule),
- and initiate retraining or firmware updates when needed.

This ensures that the system remains auditable, transparent, and overrideable, aligning with operational and regulatory requirements.

Scalability and Multi-Building Support

The edge-based design allows each building to operate as an autonomous unit. For organizations managing multiple buildings (e.g., campuses, portfolios), a lightweight orchestration layer can aggregate summary statistics or policy updates across sites. Importantly, no raw sensor data leaves the building unless explicitly authorized, and all learning remains local.

This decentralized design reduces cloud costs, avoids data privacy compliance risks, and ensures scalability—NEST instances can be cloned or adapted to other buildings with minimal effort and without re-engineering centralized infrastructure.

Discussion

The NEST architecture proposes a novel integration of generative AI, data-driven modeling, and reinforcement learning, fully embedded at the building level. This section discusses its main advantages, limitations, and positioning relative to existing control strategies in the smart building domain.

Advantages over Existing Approaches

Compared to traditional Building Management Systems (BMS) and rule-based control, NEST introduces adaptability and long-term optimization without requiring human intervention or pre-defined models. Unlike Model Predictive Control (MPC), which requires extensive calibration and manual modeling, NEST relies entirely on measured data to learn the building's thermal behavior, reducing setup cost and increasing transferability.

Compared to LLM-based agents used in isolation, which are promising for prompt-based reasoning but lack consistency, determinism, and long-term memory, NEST uses the generative layer only as a temporary bootstrap. This avoids the main pitfalls of relying on stochastic outputs for safety-critical control decisions.

Reinforcement learning has been shown to outperform both static schedules and LLM heuristics in simulation. However, its deployment in real-world buildings is often limited by the risk of unsafe exploration. NEST circumvents this by training in a learned digital twin, allowing the agent to learn policies tailored to the real building while preserving safety.

Finally, by embedding all operations locally, NEST ensures privacy, resilience to connectivity issues, and reduced dependence on external cloud services, which are often undesirable in regulated or security-sensitive environments.

Limitations and Trade-Offs

NEST requires an initial period of operation with the generative agent to collect meaningful data before any model training can begin. This startup phase typically lasts several weeks and may not yield optimal energy performance during that time. However, this cost is amortized over time as the system adapts.

The quality of the digital twin is inherently tied to the quality of the input data. Poorly calibrated sensors, missing data, or irregular HVAC responses can affect the model's accuracy. Mitigation strategies such as outlier detection, redundancy in sensing, or conservative fallbacks are necessary.

RL policies may require periodic retraining to remain effective, especially in buildings with seasonal variability, changing layouts, or evolving occupancy. While retraining is automated and local, this adds maintenance overhead compared to static rule systems.

Finally, although the edge deployment avoids cloud dependence, it limits access to large-scale compute or coordination strategies (e.g., federated learning, cross-site optimization). Extending NEST into a multi-agent learning framework across buildings is a potential avenue for future work.

Potential Extensions

Several directions can further enhance the capabilities of NEST:

- Multi-objective reward functions that include carbon intensity, occupant feedback, or electricity price signals.
- Hybrid twin models combining grey-box thermal physics with neural approximators for better generalization.
- Federated learning between buildings with similar typologies, preserving privacy while sharing policy improvements.
- Integration with demand response strategies and grid-interactive efficient buildings (GEBs).
- Deployment in residential or educational buildings with variable occupancy and comfort needs.

By treating the building as a living, learning system, NEST lays the groundwork for autonomous, adaptive energy infrastructure at scale.

Projected Energy Savings Based on Prior Work

Although this paper focuses on the architectural and algorithmic design of NEST rather than experimental deployment, several recent studies provide quantitative benchmarks that help estimate the potential energy savings achievable through each phase of the proposed pipeline.

Large language models (LLMs), when used as generative controllers for HVAC systems, have demonstrated energy reductions in the range of **15–20%** compared to static rule-based setpoints. Ahn et al. (2023) reported a **16.8%** HVAC energy saving using ChatGPT to control an EnergyPlus building model, without any task-specific training.

Deep reinforcement learning (DRL) agents trained within simulated environments—particularly those calibrated via digital twins—have achieved **20–30%** HVAC energy savings while maintaining comfort constraints. The Empa NEST project in Switzerland, for example, reported **6–40%** savings depending on season and building zone.

By combining these two approaches, NEST offers a continuous learning pipeline: LLMs provide immediate control, a data-driven digital twin enables safe simulation, and DRL policies are refined over time. Based on these foundations, we **project theoretical energy savings up to 30%**, assuming adequate sensor coverage and approximately **4–8 weeks** of real-time data for model convergence.

These gains are consistent with the current literature and emphasize the potential of embedded, continuously learning systems to significantly improve building energy performance without compromising comfort.

Conclusion

This paper introduced NEST, an edge-native architecture designed to enable autonomous energy management in buildings through a progressive learning pipeline. The system combines generative AI, a continuously improving digital twin, and reinforcement learning to optimize HVAC operation based on real-time data and historical patterns.

Starting from a cold environment, NEST uses a generative agent to initiate intelligent control decisions. It then constructs a supervised model of the building's thermal behavior using data collected in real conditions. This model is refined over time through periodic retraining, capturing the building's evolving dynamics and seasonal variations. On top of this digital twin, a reinforcement learning agent is trained to derive control policies that minimize energy consumption while respecting comfort constraints. All operations are performed locally using embedded hardware, ensuring low-latency control, data sovereignty, and full autonomy.

By integrating perception, learning, and decision-making into a single self-contained platform, NEST allows buildings to become adaptive agents capable of understanding and optimizing their own energy behavior. This architecture opens the door to scalable deployment across diverse building types, without requiring cloud services or external supervision. Future work will explore broader applications, including multi-zone coordination, federated learning across building portfolios, and integration with renewable energy systems and grid services.

As the digital twin is periodically retrained with richer and more diverse data, the reinforcement learning agent is continuously updated in response. This feedback loop allows the control policy to evolve alongside the building's real behavior, resulting in increasingly precise and efficient operation over time.

Call for Funding: Why Industry Should Invest in NEST

Graiphic is developing NEST (Next Energy Smart Twin) the world's first edge-native, self-learning AI system for building energy optimization.
NEST transforms any building into a sovereign, adaptive energy agent capable of learning from its own behavior and improving continuously without cloud dependency.
We are now opening a Call for Funding to accelerate the deployment and industrialization of NEST across Europe and Africa.

Why Invest?

- Strategic Advantage: Gain early access to a disruptive, sovereign AI platform that merges digital twins, reinforcement learning, and edge orchestration for real-time building optimization.
- Massive Market Impact: Buildings represent 40 % of global energy consumption, NEST directly targets one of the largest levers for carbon reduction.
- Energy & ROI: Achieve 25–35 % energy savings with ROI under 24 months and comfort deviation below ± 0.3 °C.
- Sovereign and Compliant: 100 % local execution (Jetson / DGX Spark), fully GDPR and ISO 50001 compliant.
- Scalability: From single-site pilots to multi-building portfolios, NEST operates without cloud infrastructure and adapts to any BMS or HVAC system.

What We Offer

- Co-development opportunities with Graiphic's engineering team for pilot and integration projects.
- Early integration of your sensors, controllers, or hardware platforms into NEST's ONNX-based orchestration.
- Visibility in international initiatives (ADRA, Horizon Europe, ITEA4, ADEME) on sovereign AI and GreenTech.
- Shared benchmarking and open data dissemination to establish your technology as a leader in energy-efficient AI.

How to Engage

Graiphic is actively seeking:

- Equity investors ready to support the scale-up of NEST across Europe and Africa.
- Industrial sponsors willing to co-fund demonstration sites and validation pilots (public buildings, universities, smart campuses).
- Strategic partners (HVAC integrators, energy companies, equipment manufacturers) who want their technologies at the heart of autonomous, self-learning buildings.

Join us in shaping the universal cockpit for AI. Contact: funding@graiphic.io | www.graiphic.io