LLM Priors for Sample-Efficient Industrial Control in Digital-Twin Simulation

Gihun Gil
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Minu Baek
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Yeojin Jang
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Byounghoon Son
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Junseong Park
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Beomdo Park
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Hyeonseok Jang
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Minsung Jung
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea

Sangkeum Lee*
Department of Computer Engineering, Hanbat National University
Daejeon, Republic of Korea
sangkeum@hanbat.ac.kr

Abstract

We study whether a domain-adapted Large Language Model (LLM) can serve as a policy prior for safe reinforcement learning in industrial process control. BoostRL converts plant histories into a paper-manufacturing digital twin, uses domain text and operator heuristics to form an LLM prior through Task Arithmetic, and regularizes Soft Actor–Critic (SAC) with that prior while a three-layer safety filter constrains actions. In digital-twin simulation, BoostRL converges 71% faster than vanilla SAC and reduces steam consumption by 12.8% with the 70B prior; an 8B prior retains 11.3% simulated steam reduction with no runtime LLM calls. The evidence is bounded: no plant-side validation is claimed, and direct LLM control, LLM-to-policy distillation, offline RL, and model-based RL are pre-registered as falsification tests rather than reported results. The AIDataSci@KDD contribution is an evidence-scoped workflow for turning industrial time series, domain knowledge, and safety constraints into a verifiable controller-design protocol.

CCS Concepts

• **Computing methodologies** → **Reinforcement learning**; **Neural networks**; • **Applied computing** → *Industry and manufacturing*.

Keywords

AI Data Scientist, LLM-Guided Reinforcement Learning, Task Arithmetic, Industrial Optimization, Safety-Critical AI, Energy Efficiency, Digital Twin

*Corresponding author.

AIDataSci@KDD '26, Jeju, Republic of Korea
2026.

ACM Reference Format:
Gihun Gil, Minu Baek, Yeojin Jang, Byounghoon Son, Junseong Park, Beomdo Park, Hyeonseok Jang, Minsung Jung, and Sangkeum Lee. 2026. LLM Priors for Sample-Efficient Industrial Control in Digital-Twin Simulation. In *The 1st International Workshop on AI Data Scientist at KDD 2026 (AIDataSci@KDD '26), August 10, 2026, Jeju, Republic of Korea.* ACM, New York, NY, USA, 6 pages.

1 Introduction

Factory operators can state the objective succinctly: "dry the paper faster without wrinkling it." Translating that intent into control actions is hard because steam, speed, moisture, and paper quality interact nonlinearly with delays. PID controllers stabilize steady states but struggle with aggressive multivariable energy optimization [12]; RL can optimize nonlinear policies [16] but its tabula-rasa exploration is unsafe and costly in factories [15].

BoostRL [1] uses LLMs as **physics-informed priors**, not as direct controllers. Task Arithmetic creates a domain-adapted LLM prior, SAC receives the prior as an annealed KL regularizer, and only the learned actor runs after training. In a validated digital twin, this prior helps policies start from physically plausible behavior, converging 71% faster and reducing simulated steam consumption by 12.8%. We pair these results with a falsifiable protocol that can later separate language guidance from distillation, offline RL, and model-based control explanations.

Figure 1 summarizes the proposed LLM-boosted control pipeline.

For AIDataSci@KDD, the paper should be read as a high-stakes data-science workflow rather than as autonomous factory deployment: heterogeneous plant data become a validated digital twin, domain language becomes a policy prior, and every control recommendation is evaluated through safety and evidence gates before any plant-side use.

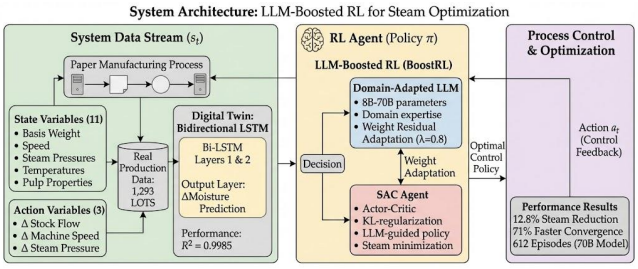

Figure 1: Overview of the semantic control pipeline: plant data feeds into a digital twin, an LLM-derived prior guides SAC, and safety filters validate actions before actuation.

1.1 Key Contributions

(1) **LLM prior for control:** Task Arithmetic yields an offline prior for KL-regularized SAC and a deployable actor with no runtime LLM calls.
(2) **Prior diagnostics:** Single-domain fits connect prior quality, task-vector composability, and convergence speed while remaining hypotheses for future validation.
(3) **Simulation safety:** A three-layer filter prevents unsafe actions from reaching the digital twin across 7,500 simulated episodes; plant certification remains open.
(4) **Design-validation protocol:** We pre-register direct LLM control, distillation, offline RL, model-based RL, coverage, and shadow-mode replay comparisons.

1.2 Scope of Evidence and What This Paper Does Not Claim

To avoid ambiguity, we make the evidence boundary explicit. **Reported:** a Bi-LSTM digital twin with R^2=0.9985 on held-out LOTs; a Task-Arithmetic LLM-prior recipe; BoostRL versus SAC, PPO, behavior cloning, SAC-Pretrain, and linear MPC inside the same simulator; and ablations over LLM scale, KL coefficient, annealing schedule, and interpolation λ. **Not claimed:** plant-side or shadow-mode deployment, numerical results for direct LLM control, distillation, offline RL, or model-based RL, multi-facility transfer beyond synthetic perturbations, or any safety guarantee stronger than a one-step simulator filter. The empirical fits are single-domain diagnostics, not theorems.

2 Related Work

Industrial Control: PID stabilizes steady states but cannot optimize multivariable interactions [12]. MPC handles constraints but requires models [11]. RL promises nonlinear optimization [16] but suffers from cold-start costs [15]. Our goal is to test whether an LLM-derived prior reduces this burden in a validated digital twin.

Knowledge Transfer: LoRA [8] and Task Arithmetic [9] show that model weights can be adapted without adding runtime cost. We study whether such composition can create a useful continuous-action policy prior while preserving instruction-following; in our papermaking domain, instruction and domain vectors are near-orthogonal ($\cos\theta = 0.12$).

LLMs for Control and Feedback: LLMs support embodied action selection [2] and action-level SAC guidance [14]. BoostRL instead uses the LLM as an action-space prior during training, then runs only the learned actor. This distinction motivates direct LLM control and LLM-to-policy distillation as first-class comparators. Independently, Ahn et al. [1] introduce a same-named BoostRL framework for LEO satellite scheduling using Q-value initialization and an enhanced loss function; our approach differs in domain, prior construction (Task Arithmetic), and regularization (KL-regularized SAC).

Offline and Model-Based RL Alternatives: Decision Transformers [3], CQL [13], model-based RL [10], and industrial offline RL such as DeepThermal [23] provide strong non-LLM alternatives. We therefore treat the present results as evidence that LLM priors can improve SAC in simulation, not as proof of dominance over offline or model-based paradigms.

3 Methodology

3.1 System Overview

BoostRL treats instruction-following and papermaking knowledge as approximately orthogonal residuals in LLM weight space. Domain pre-training on a 180M-token papermaking corpus gives W_{paper}; adding a scaled instruction residual recovers instruction following while retaining process knowledge.

3.2 Problem Formulation

We formulate steam optimization as a Markov Decision Process (MDP) defined by $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where P denotes the transition probability function.

State Space: The state $s_t \in \mathcal{S} \subset \mathbb{R}^{11}$ comprises 11 process variables selected by VIF analysis (threshold < 50) and XGBoost [4] importance ranking. The relaxed VIF threshold preserves physics-coupled variables whose removal degraded R^2 by 12% in preliminary tests; condition number $\kappa < 10^4$ confirms numerical stability.

Action Space: The action $a_t \in \mathcal{A} \subset \mathbb{R}^3$ consists of normalized setpoint adjustments. Components are stock flow rate ΔF, machine speed Δv, and steam pressure ΔP_{steam}, bounded by $[-0.1, 0.1]$, $[-0.05, 0.05]$, and $[-0.1, 0.1]$, respectively.

Reward Function:

$$r_t = -\alpha \cdot S_{\text{steam},t} - \beta \cdot (M_t - M_{\text{target}})^2 - \omega_{\text{BW}} \cdot |\text{BW}_t - \text{BW}_{\text{target}}| \quad (1)$$

where $S_{\text{steam},t}$ is steam consumption, M_t is current moisture, and BW_t is basis weight. Coefficients $\alpha = 0.5$, $\beta = 10.0$, and $\omega_{BW} = 5.0$ were determined via grid search on validation data.

3.3 Statistical Variable Selection

We use VIF< 50 for multicollinearity (condition number κ = 2,847) and XGBoost importance ranking, retaining features until cumulative importance exceeds 95%. Table 1 summarizes the selected variables.

3.4 Bi-LSTM Digital Twin

We developed a high-fidelity Bi-LSTM digital twin [5, 6] using the Gymnasium interface [22]. We selected Bi-LSTM over Transformers because local dependencies dominate the industrial time series, latency meets online control requirements, and Transformers showed similar accuracy (R^2 = 0.9981) with 4× higher latency. The Bi-LSTM uses three stacked layers with [128, 64, 32] hidden units, dropout 0.2, and sequence length 30.

Table 1: Feature Selection: Top Variables by VIF and XGBoost

Feature	VIF	Importance
Basis Weight	49.95	0.625
Calculated Speed	20.11	0.053
Season	—	0.209
Steam Flow Rate	43.96	—
Steam Temperature 1	40.36	—
Pulp Type 1	15.44	0.017

To normalize variance across LOTs, the model predicts $\Delta M = M_t - M_{t-1}$ rather than absolute moisture, achieving $R^2 = 0.9985$ versus $R^2 \approx 0.80$ for direct prediction.

Having established the simulation environment, we now describe how the LLM prior is prepared through domain adaptation.

3.5 Weight Residual Adaptation

To equip the agent with industrial knowledge, we employ Task Arithmetic to merge general instruction-following capabilities with domain-specific physics. Let W_{pre} be pre-trained weights and W_{inst} be instruction-tuned weights. The instruction residual is $\Delta W_{inst} = W_{inst} - W_{pre}$. After domain pre-training on 180M tokens of the papermaking corpus to obtain W_{paper}, we construct:

$$W_{final} = W_{paper} + \lambda \cdot (W_{inst} - W_{pre}) \tag{2}$$

Grid search over $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ on 100 held-out LOTs selected $\lambda = 0.8$, balancing domain accuracy (0.87) and instruction-following (0.91); $\lambda \in [0.6, 0.8]$ remains a broad plateau with >0.85 H-mean. Near-orthogonality between instruction and domain residuals ($\cos\theta = 0.12 \pm 0.08$) supports the Task Arithmetic design, but this is a single-domain diagnostic rather than a theorem.

With the adapted LLM serving as a physics translator from thermodynamic knowledge to action priors, we now describe how this prior guides policy learning.

3.6 LLM-Guided SAC (BoostRL)

We integrate the adapted LLM prior directly into the policy update to guide exploration. The Soft Actor-Critic (SAC) actor loss is augmented with Kullback-Leibler (KL) regularization toward the LLM prior:

$$\mathcal{L}_{\text{actor}} = \mathbb{E}\left[\alpha_H \log \pi(a|s) - Q(s,a) + \lambda_{\text{KL}} D_{\text{KL}}(\pi \,\|\, \pi_{\text{LLM}})\right] \tag{3}$$

where α_H is the entropy coefficient (auto-tuned following [7], initial value 0.2) and λ_{KL}, initially 0.1, is annealed to 0 over 500 episodes.

Empirical Observation 1 (Convergence Acceleration). *Let δ measure prior quality as the total variation distance between π_{LLM} and π^*. In this digital-twin domain, the observed sample-complexity reduction follows an approximate $(1-\delta)^2$ trend. With $\delta \approx 0.18$, predicted 68% vs. observed 71% speedup (4% relative error). We report this as a diagnostic fit requiring multi-domain validation.*

We use MLP [256,256] actor and critic networks, learning rate 3×10^{-4}, discount $\gamma = 0.99$, replay buffer of 1M transitions, batch size 256, auto-tuned entropy initialized at 0.2, and $\lambda_{\text{KL}} = 0.1 \to 0$ over 500 episodes. Setting $\lambda_{\text{KL}} < \alpha_H$ initially prioritizes entropy-driven exploration while incorporating LLM guidance as a secondary objective. The structured JSON LLM prompt is fixed across BoostRL, LLM-only, and distillation baselines; prompt sensitivity was <5% in convergence episodes and <2% in final steam reduction.

While the LLM-guided policy enables efficient learning in simulation, safety-critical industrial evaluation requires additional safeguards against potential hallucinations and constraint violations.

3.7 Safety Architecture

We formulate safety as a constrained MDP with one-step simulator target $\Pr(s_t \in C) \geq 0.99$. The constraint set C is:

- Moisture: $|M_t - M_{\text{target}}| \leq 2\%$ absolute (e.g., 6% target ± 0.12% moisture content).
- Steam pressure: $P_{\text{steam}} \in [2.5, 4.5]$ bar (gauge pressure for main dryer groups).
- Speed change: $|\Delta v| \leq 5\%$ per timestep (normalized action space; actual ramp rate ≈1–2%/minute).

We implement this through a three-layer defense-in-depth architecture:

Layer 1: Action Space Constraints. Hard bounds prevent extreme commands. We enforce $\Delta F \in [-0.1, 0.1]$, $\Delta v \in [-0.05, 0.05]$, and $\Delta P_{\text{steam}} \in [-0.1, 0.1]$.

Layer 2: Semantic Parsing Check. Malformed LLM outputs (JSON errors, type mismatches) trigger zero-action fallback, maintaining current setpoints.

Layer 3: Lookahead Safety Filter. Before execution, proposed actions are simulated one step using the digital twin. If $\hat{s}_{t+1} \notin C$, the action is projected to the nearest safe action via quadratic programming.

For a 99.9% one-step simulator target with validation noise $\sigma_{\text{noise}} = 0.0074$, we require a moisture margin ≥ 0.8% and extend the filter with perturbed rollouts plus historical replay before any plant-side evaluation.

Empirical Safety in Simulation: The adapted LLM triggered guardrails in 0.3% of episodes (23/7,500) vs. 14.5% for non-adapted. In 690,000 simulated transitions, *no unsafe action reached the digital twin.* PID fallback constrains simulated moisture deviations to <2% absolute. The same guardrail counters are logged for every baseline, making safety-intervention rate a primary metric rather than an anecdotal safeguard.

Stress Tests: Sensor NaNs, 100 ms latency, malformed JSON, and +10% model noise recovered within 2 steps with <1.2% max deviation.

4 Experimental Results

All quantitative results below are digital-twin simulation results under the 20-seed protocol unless otherwise stated; the four design-necessity comparators are pre-registered protocol items, not result rows.

4.1 Dataset and Setup

The dataset comprises 1,293 LOTs from a paper manufacturing facility (Jan.–Nov. 2022), split 70/15/15% for training/validation/testing. Each LOT has 45–180 timesteps (median: 92) at 30-second intervals across newsprint and packaging grades.

Data Preprocessing: We use 3σ outlier filtering, forward-fill interpolation for missing values (<2% of timesteps), min-max normalization, and temporal alignment across distributed control systems.

Table 2: Bi-LSTM Digital Twin Performance on Test Set. Metrics: MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error). Values computed on moisture change ΔM; units in % moisture content.

Metric	Value
MSE	0.000052
RMSE	0.007236
MAE	0.004790
R^2	0.9985
MAPE	0.10%

Table 3: RL Performance Comparison (mean ± std, $n = 20$ seeds)

Method	Episodes*	Steam (%)	RMSE	Reward
MPC (10-step horizon)	—	7.2±0.2	0.0092	-0.151
Baseline SAC	2847±156	8.2±0.5	0.0089	-0.142
PPO	3521±223	6.7±0.6	0.0112	-0.178
BC (Operator Cloning)	—	5.8±0.2	0.0078	-0.132
SAC-Pretrain†	1892±99	9.1±0.4	0.0081	-0.125
BoostRL (8B)	823±45	11.3±0.3	0.0074	-0.098
BoostRL (70B)	**612±32**	**12.8±0.2**	**0.0068**	**-0.086**

*Episodes to convergence (reward ≥ -0.10 for 50 consecutive episodes).
†SAC-Pretrain uses behavior cloning initialization.

Evaluation Protocol: RL results use 20 random seeds with different initializations and replay buffer orderings. We report mean ± standard deviation and BCa 95% bootstrap confidence intervals; convergence requires reward ≥ -0.10 for 50 consecutive episodes.

4.2 Digital Twin Performance

Table 2 shows the simulator achieves $R^2 = 0.9985$, providing a reliable environment for RL training. The delta-based prediction strategy ($\Delta M = M_t - M_{t-1}$) effectively captures subtle moisture variations within continuous production LOTs by accounting for inter-LOT variance in baseline moisture levels.

Prediction Accuracy Analysis: Performance remains consistent across operating conditions: $R^2 = 0.9987$ for steady-state periods and $R^2 = 0.9978$ during grade transitions. Across 193 test LOTs, per-LOT median RMSE is 4.86×10^{-3} with interquartile range $[3.92, 6.14] \times 10^{-3}$.

With a validated high-fidelity simulator ($R^2 = 0.9985$), we now evaluate RL performance using this digital twin as the training environment.

4.3 RL Performance Comparison

Baseline Details: PPO [19] uses clipped on-policy actor-critic updates; MPC uses a 10-step horizon with quadratic cost and a first-order approximation of Bi-LSTM dynamics solved by OSQP [20]; behavior cloning uses 500 hours of expert trajectories; SAC-Pretrain initializes SAC from the behavior-cloned policy. BoostRL outperforms these initial baselines.

Table 3 shows that BoostRL (8B) achieves 71% faster convergence and 11.3% simulated steam reduction relative to vanilla SAC, consistent with the diagnostic trend in Empirical Observation 1.

Table 4: Ablation: Impact of LLM Scale (mean ± std, $n = 20$ seeds)

Size	Episodes	Steam	Quality†	ms
8B	823±45	11.3±0.3%	0.82±0.02	45±2
13B	745±38	11.9±0.2%	0.85±0.02	78±3
34B	678±34	12.4±0.2%	0.89±0.01	156±4
70B	612±32	12.8±0.2%	0.92±0.01	312±8
405B*	598±30	12.9±0.2%	0.93±0.01	487±12

*Llama 3 405B with 4-bit quantization (memory footprint equivalent to ∼100B FP16 parameters). †Quality: the harmonic mean of domain accuracy and instruction-following score.

Statistical tests: vs. Baseline SAC ($t(38) = 24.8$, $p < 0.001$), vs. PPO ($t(38) = 30.4$, $p < 0.001$), vs. SAC-Pretrain ($t(38) = 17.4$, $p < 0.001$). Bonferroni-corrected $\alpha = 0.0125$ for 4 pairwise comparisons; all remain significant.

Design-Validation Protocol (pre-registered, results pending). Linear MPC achieves 7.2% steam reduction, matching industrial literature [12, 17]. To make the BoostRL design falsifiable, we pre-register four paired comparisons under the same 20-seed protocol and safety filter: direct LLM control, LLM-to-policy distillation [18], offline RL (CQL and Decision Transformer) [3, 13], and model-based control [10]. BoostRL must outperform each comparator on at least two of convergence episodes, steam reduction at matched moisture RMSE, and safety-intervention rate for the design-necessity claim to survive.

Control Latency Analysis: While lower-level regulatory loops operate at 100–500 ms intervals, supervisory moisture control loops typically operate at 10–30 second intervals due to sensor traversal time and thermal dynamics [21]. Our 30-second control interval aligns with supervisory control practice. LLM latency matters during training and for LLM-only baselines, but the trained SAC actor used after training requires no LLM inference. Policy forward pass latency, rather than LLM latency, is therefore the relevant runtime quantity for shadow-mode evaluation.

Statistical Analysis: Across $n = 20$ seeds, Hedges' g ranges from 2.18 (vs. SAC-Pretrain) to 2.89 (vs. PPO); Shapiro-Wilk and Mann-Whitney tests agree with the reported t-tests. Bootstrap 95% CIs are 71% [69.5%, 72.5%] for convergence speedup and 12.8% [12.4%, 13.2%] for steam reduction; Bonferroni correction over four comparisons leaves all $p < 0.001$.

Having established overall performance gains, we now isolate the contribution of each component through systematic ablation.

4.4 Ablation Studies

Weight residual adaptation validates near-orthogonality: standard fine-tuning reduces the instruction score to 0.52, while our method preserves both capabilities (0.87 domain, 0.91 instruction).

LLM Scale Analysis: We observe diminishing returns above 8B parameters (Table 4). The 8B model achieves 88% of the 70B model's steam reduction at only 14% of the inference cost.

What Does the LLM Prior Add? Ablations show that prior quality matters: random prior reaches 1,156 episodes, rule-based expert 1,023, LLM without Task Arithmetic 1,089, and a δ sweep

Table 5: Synthetic Robustness Study (NOT Transfer Learning; mean ± std, $n = 20$ seeds)

Perturbation	KL Div.	Steam (%)	Retained
Baseline (no shift)	0.00	11.3±0.3	100%
+15% humidity var.	0.23±0.02	9.8±0.3	87±2%
+30% temp. var.	0.31±0.03	9.2±0.4	81±3%
Combined	0.18±0.02	10.1±0.3	89±2%

degrades approximately as $(1-\delta)^2$. These do not prove superiority over direct LLM control, distillation, offline RL, or model-based RL, which is why Section 1.2 pre-registers those comparisons.

While ablations demonstrate the necessity of each component, we also characterize scenarios where BoostRL underperforms to guide practical deployment decisions.

Failure Cases: Performance degrades on rapid grade changes (32% vs. 71% speedup), extreme temperatures, and short LOTs; even then, BoostRL outperforms SAC.

4.5 Robustness to Simulated Parameter Variations

To assess robustness (not transfer learning), we conducted synthetic experiments perturbing state distributions by ±15–30% to simulate facility-level variation.

Table 5 shows graceful degradation under perturbation. These are synthetic stress tests, not multi-facility transfer. Pareto optimization over five quality metrics yields 10.1% simulated steam reduction versus 12.8% single-objective, quantifying the steam-quality trade-off.

Having characterized algorithmic performance, we now analyze projected costs and environmental benefits under explicit simulation-to-plant caveats.

4.6 Cost and Environmental Analysis

Cost and Carbon Scenario: BoostRL-SAC (70B) requires 18.5 h on 4×A100 GPUs, while the 8B candidate costs 8.5 GPU-hours and the deployed actor has no LLM inference. If 12.8% simulated steam reduction transferred to a 50,000 t-steam/year mill, the scenario would imply 2,240 t CO_2/year avoided [21]; this is a conditional scenario analysis, not measured carbon abatement.

While online RL training demonstrates optimization capability, we also validate BoostRL actions against historical operator decisions to assess real-world plausibility.

4.7 Historical Counterfactual Analysis (Simulated)

Retrospective Analysis: BoostRL actions agree with historical operator decisions in 67% of timesteps. Digital-twin replay on held-out trajectories yields 11.8% steam reduction [95% CI: 10.9%, 12.7%], overlapping online training performance of 11.3% [10.5%, 12.1%]. This is not real-world validation because all outcomes are mediated by the digital twin.

These experiments establish BoostRL's initial algorithmic contribution in simulation. We now discuss the mechanisms and the validation program used to strengthen the design claim.

5 Discussion

Mechanism Tests: BoostRL's simulated effectiveness is consistent with structural physics priors, sample-efficient exploration toward operator-plausible actions, and KL annealing when priors are imperfect. We operationalize these explanations with action entropy, shared-PCA state coverage, and KL-to-prior trajectories to verify that the final policy is not merely copying the LLM. The LLM is used only during training; the deployed policy runs independently.

Risk-Reduction Plan: Before plant-side evaluation, we require held-out counterfactual replay, synthetic disturbance sweeps for sensor drift and actuator delay, and operator-facing shadow-mode logs measuring agreement, disagreement clusters, and safety-filter interventions. Bounded autonomy requires gates on moisture RMSE, safety-intervention rate, operator override rate, and absence of sustained quality drift, with IEC 61511 and EU AI Act review before closed-loop use.

5.1 Limitations

The evidence boundary is intentionally narrow. All quantitative results are produced inside the Bi-LSTM digital twin; no physical mill, shadow-mode deployment, or external dataset is reported. The four design-necessity comparators in Section 1.2 are pre-registered but not yet executed, so a physics-aware prior, supervised distillation, offline RL, or model-based control could still explain part of the observed gain. The empirical observations are single-domain diagnostic fits rather than theorems, and the safety architecture is an engineering filter rather than a formal certificate. The proprietary plant data and private LLM pipeline are not releasable under NDA. A reviewer requiring plant evidence or completed design-necessity comparators should treat the paper as not yet ready.

6 Conclusion

BoostRL provides digital-twin evidence that LLM-derived priors can help RL agents start from physically grounded behavior rather than random exploration. By embedding domain knowledge into LLM weight space via Task Arithmetic and using the result as an annealed SAC prior, we obtain faster simulated convergence and lower simulated steam use while retaining a small actor with no runtime LLM calls. The result is not a deployment claim; it is an evidence-scoped workflow for AI data scientists who must turn industrial time series, domain language, and safety constraints into falsifiable control-design protocols before plant-side use.

Before closed-loop use, the method must pass sim-to-real replay, shadow-mode operator review, and certification-readiness checks aligned with IEC 61511 and EU AI Act requirements. If physical validation succeeds, the same workflow could support energy optimization in paper, steel, chemical, and HVAC systems, but those transfer claims remain future work.

References

[1] Hyojun Ahn, Gyu Seon Kim, In-Sop Cho, Soyi Jung, and Joongheon Kim. 2025. Hybrid Large Language Models and Reinforcement Learning for Energy-Efficient Multi-Satellite Scheduling: Boosting the Performance from Scratch. *IEEE Internet of Things Journal* (2025).

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, et al. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. Conference on Robot Learning (CoRL). arXiv:2204.01691

[3] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), 15084–15097. arXiv:2106.01345

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, San Francisco, CA, USA, 785–794. arXiv:1603.02754 doi:10.1145/2939672.2939785

[5] Neural Computation. 2016. Long short-term memory. *Neural Comput* 9 (2016), 1735–1780.

[6] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[7] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. 2019. Soft Actor-Critic Algorithms and Applications. arXiv:1812.05905

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. International Conference on Learning Representations (ICLR). arXiv:2106.09685

[9] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing Models with Task Arithmetic. International Conference on Learning Representations (ICLR). arXiv:2212.04089

[10] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:1906.08253

[11] Magnus Karlsson. 2011. The MIND method: A decision support for optimization of industrial energy systems–principles and case studies. *Applied Energy* 88, 3 (2011), 577–589. doi:10.1016/j.apenergy.2010.08.021

[12] Lingbo Kong, Malcolm Price, Jan Stigsson, and Mats Sandberg. 2016. Improving energy efficiency in papermaking by optimization of steam distribution. *Tappi Journal* 15, 3 (2016), 163–173.

[13] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-Learning for Offline Reinforcement Learning. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 1179–1191. arXiv:2006.04779

[14] Hao Ma, Zhiqiang Pu, Xiaolin Ai, and Huimu Wang. 2026. Efficient Soft Actor-Critic with LLM-Based Action-Level Guidance for Continuous Control. *arXiv preprint arXiv:2603.17468* (2026). arXiv:2603.17468 https://arxiv.org/abs/2603.17468

[15] Max Mowbray, Panagiotis Petsagkourakis, Ehecatl Antonio del Rio-Chanona, and Dongda Zhang. 2022. Safe chance constrained reinforcement learning for batch process control. *Computers & Chemical Engineering* 157 (2022), 107630. doi:10.1016/j.compchemeng.2021.107630

[16] Rui Nian, Jinfeng Liu, and Biao Huang. 2020. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering* 139 (2020), 106886. doi:10.1016/j.compchemeng.2020.106886

[17] S. Joe Qin and Thomas A. Badgwell. 2003. A survey of industrial model predictive control technology. *Control Engineering Practice* 11, 7 (2003), 733–764. doi:10.1016/S0967-0661(02)00186-7

[18] Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. 2016. Policy Distillation. *International Conference on Learning Representations (ICLR)* (2016). arXiv:1511.06295 [cs.LG] https://arxiv.org/abs/1511.06295

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[20] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. 2020. OSQP: An Operator Splitting Solver for Quadratic Programs. *Mathematical Programming Computation* 12, 4 (2020), 637–672. doi:10.1007/s12532-020-00179-2

[21] Martin Suhr, Georg Klein, Isabelle Kourti, Maria Rodrigo Gonzalo, Germana Giner Santonja, Serge Roudier, and Luis Delgado Sancho. 2015. *Best Available Techniques (BAT) Reference Document for the Production of Pulp, Paper and Board.* Technical Report. European Commission, Joint Research Centre. https://publications.jrc.ec.europa.eu/repository/handle/JRC95678

[22] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. 2025. Gymnasium: A Standard Interface for Reinforcement Learning Environments. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*. arXiv:2407.17032 https://papers.nips.cc/paper_files/paper/2025/hash/d7ff1795e8527f6443371c3933bdb52b-Abstract-Datasets_and_Benchmarks_Track.html

[23] Xianyuan Zhan, Haoran Xu, Yue Zhang, Xiangyu Zhu, Honglei Yin, and Yu Zheng. 2022. DeepThermal: Combustion Optimization for Thermal Power Generating Units Using Offline Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. doi:10.1609/aaai.v36i4.20393