ACTIVE SUPERVISION FOR AGS BUNCH-MERGING WITH LLM-BASED REINFORCEMENT LEARNING*

Y. Zhao[†1], K. Brown[2], A. Edelen[3], Y. Gao[2], E. Hamwi[4], G. H. Hoffstaetter[4], A. Kasparian[5], D. Kuzovkova[4], W. Lin[2], T. Miceli[6], J. Morris[2], M. Schram[5], V. Schoefer[2], A. Sukhanov[2], S. Tajne[2], J. Unger[4], K. Zeno[2], Y. Wang[‡1]
[1]Industrial and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY, USA
[2]Collider-Accelerator Department, Brookhaven National Laboratory, Upton, NY, USA
[3]SLAC National Accelerator Laboratory, Menlo Park, CA, USA
[4]CLASSE, Cornell University, Ithaca, NY, USA
[5]Thomas Jefferson National Accelerator Facility, Newport News, VA, USA
[6]Fermi National Accelerator Laboratory, Batavia, IL, USA

Abstract

Radio-frequency (RF) bunch-merging gymnastics is used in the RHIC heavy-ion program to combine individual source pulses into single bunches with suitable intensity. Preserving intensity and emittance during these gymnastics requires careful coordination of the voltages and phases of RF cavities at several harmonic numbers, which is labor-intensive and fragile against machine drift. Recent work using a physics-based simulator of the Brookhaven Alternating Gradient Synchrotron (AGS) has shown that reinforcement learning (RL) can learn effective merge configurations. RL is data-intensive and requires many training interactions with the environment. Large language models (LLMs) have recently demonstrated the ability to extract patterns from large, noisy data and to integrate domain knowledge into the control loop, making them an attractive aid for tuning complex accelerator systems. However, domain adaptation (i.e., prompt engineering, finetuning, etc.) is always required for deploying LLM in the target domain and has not been investigated in particle accelerators. To fill this gap, we propose an active supervision framework in which the LLM-based teacher first transfers general control principles from human operators to the student agent. Then, the student agent further finetunes the control policy by interacting with the simulator/experiments with improved sample efficiency.

INTRODUCTION

The Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory (BNL) provides the world's only high-energy polarized proton beam and is therefore in a unique position to study the spin structure of the nucleon [1, 2]. The polarized proton beam is generated by an Optically Pumped Polarized Ion Source (OPPIS) and then accelerated through the Linac, Booster, and Alternating Gradient Synchrotron (AGS) before injection into RHIC [3, 4].

* This work was supported in part by the grants DE-SC0024287 and DE-SC0025351.
† zhaoy23@rpi.edu
‡ wangy88@rpi.edu

Preserving polarization along this chain is a difficult task because the spin of each particle evolves slightly differently as it samples different fields along its trajectory, leading to depolarization at numerous spin resonances [5,6]. To reduce space charge effects and improve the beam polarization, the beam bunches are split in the Booster into two or four smaller bunches. The smaller bunches with reduced space charge will limit polarization loss during particle acceleration [7,8].

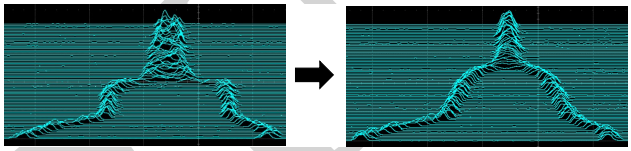

Figure 1: Mountain-range wall-current-monitor (WCM) signal before (left) and after (right) RF bunch-merging gymnastics in the AGS [9].

Before transfer to RHIC, the lower-intensity, smaller bunches are recombined into a single high-intensity bunch with suitable intensity near the extraction energy of AGS. This step is known as radio-frequency (RF) bunch-merging gymnastics. It is essential for preserving both intensity and emittance. As shown in Fig. 1, the mountain-range wall-current-monitor (WCM) signal evolves from multiple separated bunches into a single merged bunch after correction.

At present, the AGS bunch-merging process relies heavily on expert interpretation of mountain range WCM measurements. Experts evaluate the evolution of bunch widths and center-of-charge oscillations, and then iteratively adjust the RF voltage and phase programs. While this procedure can produce acceptable merges, it is labor-intensive, difficult to reproduce, and fragile against machine drift. Maintaining high merge quality requires constant monitoring and retuning, which motivates machine learning methods that can learn robust RF settings and sustain consistent performance during the AGS bunch-merging process.

Recent work has shown that reinforcement learning (RL) can provide an effective framework for the AGS bunch-merging process. Gao *et al.* [10] used an inverse-reinforcement-learning (IRL) approach on a Bmad-based simulator and showed that expert merge behavior can be learned as an RF control policy. In addition, Gao *et al.* [9] de-

ployed a Soft Actor-Critic agent on the AGS bunch-merging process and demonstrated that the agent effectively corrected the bunch merge. Together, these results confirm that RL is a promising tool for bunch merging.

However, RL is highly data-intensive and requires many training interactions with the environment, which becomes a critical bottleneck as the number of RF control points grows [11, 12]. This issue is especially important for RF bunch merging, where each additional RF voltage or phase control point increases the search space. Recent advances in large language models (LLMs) have demonstrated their ability to extract patterns from large, noisy data and adapt to new tasks with minimal supervision [13, 14], which has been applied in accelerator applications [15, 16]. However, directly using a large LLM as the controller for RF bunch merging is impractical because repeated LLM inference can introduce latency and computational overhead. A more suitable strategy is to use the LLM as a teacher during training rather than as the deployed controller. Zhou *et al.* introduced this framework, where an LLM-based teacher provides action guidance to a smaller and specialized RL student agent [17]. The student distills prior knowledge from the teacher's guidance and then continues to improve through feedback from the environment. This teacher-student structure can reduce unnecessary exploration during RL training while retaining the fast inference of a compact RL student policy.

This LLM-based policy-teacher strategy has not yet been investigated for accelerator control applications. To fill this gap, we propose a framework that trains a smaller, specialized student RL agent using instructions from an LLM-based teacher agent. The teacher agent is used only during training to provide prior knowledge of RF bunch-merging gymnastics, retrieved examples of successful merges, and action-space constraints. The student agent then refines its policy through interaction with the AGS merge simulator and becomes the deployable controller. We demonstrate that this framework provides a warm start for the student agent, leading to faster convergence and stronger early performance than RL trained from scratch.

METHODOLOGY

The proposed framework consists of two coupled components: an LLM-based teacher agent and a lightweight student RL agent. The overall architecture is shown in Fig. 2. The Environment receives RF voltage actions. The teacher agent observes the current beam state together with retrieved examples of successful historical merge actions, and then proposes RF voltage actions and returns beam quality as a reward. The student agent interacts directly with the environment and is trained with a standard RL paradigm (i.e., Soft Actor Critic [18]) augmented by guidance from the LLM teacher. Thus, the LLM is used only during training to provide active supervision, while the compact student policy becomes the deployable controller.

This framework uses an LLM-based teacher agent to transfer prior knowledge to the student RL agent. The teacher provides RF-control prior knowledge, few-shot examples of successful merges, and action-space constraints. The student policy specializes in the AGS merge process through feedback from the Environment. Compared with RL trained from scratch, the proposed framework is intended to provide a warm start, reduce unnecessary exploration, and improve early training performance.

Teacher Agent with Prompt Engineering

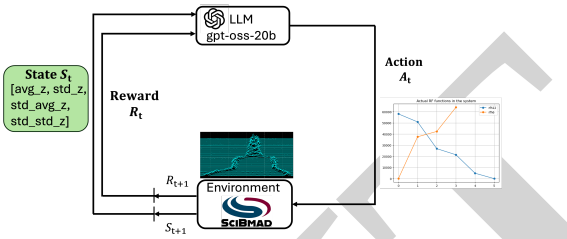

Figure 2: Architecture of the LLM-based teacher agent. Given the current beam state S_t, the LLM proposes an RF voltage action that is applied to the Environment. The Environment returns the reward R_t.

The static prompt template is organized into five blocks. The role description block establishes the LLM as an expert RF control engineer and beam dynamics specialist, and frames the task as a contextual bandit problem. The background knowledge block encodes operator know-how about the bunch-merging process. The input description block specifies the observation variables that constitute the beam state. The objective and action space block defines the optimization target and the dimension of the RF voltage action. The hard limits block enforces physically valid voltage ranges as constraints that the LLM output must satisfy.

To inject experience-driven priors, we use retrieval augmented prompting. The historical merge actions are maintained, and at each query, the top-k examples with the highest reward are retrieved and inserted into the prompt as few-shot demonstrations. This concept makes the teacher generalize from qualitatively similar past merges without any update to the LLM parameters.

Student RL Agent and Training Objective

The student policy $\pi_\theta(\cdot \mid s)$ is a lightweight model that operates over the same RF voltage action space as the teacher. Unlike the teacher, the student is updated by interacting with the environment and observing rewards, so it can refine the teacher agent's guidance with feedback from the environment. To combine the two information sources, the student is trained with the composite objective

$$\mathscr{L}(\theta) = \mathscr{L}_{\mathrm{RL}}(\theta) + \lambda\, \mathbb{E}_{s\sim\pi_\theta} \|A_T(s) - A_\theta(s)\|_2^2. \quad (1)$$

The first term $\mathscr{L}_{\mathrm{RL}}(\theta)$ is the standard RL loss that uses feedback from the environment (i.e., Soft Actor Critic [18]) and is designed to maximize the expected return obtained by the student agent. The second term is a distillation loss that measures the mean-squared error (MSE) between the

teacher's proposed RF voltage action $A_T(s)$ and the student's proposed RF voltage action $A_\theta(s)$ [19]. The hyperparameter $\lambda \geq 0$ controls the extent to which the student relies on the teacher. When $\lambda = 0$, the objective reduces to standard RL training with no teacher influence. As λ increases, the student is more strongly influenced by the teacher's guidance.

SIMULATION STUDY

We evaluate the proposed framework on the polarized proton merges using a Bmad simulation environment [20]. The action consists of six RF voltage setpoints that reconstruct the two RF voltage programs: four setpoints for harmonic group 12 and the remaining two for harmonic group 6. The state S_t returned by the environment is the beam-quality observables, S_t = [avg_z, std_z, std_avg_z, std_std_z], where avg_z is the average bunch centroid position, std_z is the average bunch length, std_avg_z is the standard deviation of the centroid, and std_std_z is the standard deviation of the bunch length.

The reward R_t is a weighted sum of three bunch-merge quality metrics:$R_t = \frac{1}{2}r_1 + \frac{1}{4}(r_2 + r_3)$, where $r_1 = (10 - \text{std\_z})/10$, $r_2 = (0.03 - \text{std\_avg\_z})/0.03$, and $r_3 = (0.3 - \text{std\_std\_z})/0.3$. Here, std_z represents the average bunch length, std_avg_z represents the centroid-position variation, and std_std_z represents the bunch-length variation. so that a higher reward corresponds to RF voltages that produce a compact and stable merged bunch, which is suitable for preserving the required final bunch intensity while limiting longitudinal emittance growth.

The teacher policy is implemented with the gpt-oss-20b LLM [21], queried with the structured prompt. At each query, the top-100 historical merge actions ranked by reward are retrieved from the JSON dataset and inserted into the prompt as few-shot demonstrations, following the retrieval-augmented generation paradigm [22]. The student is a lightweight model, trained with the composite objective in Eq. (1). Deploying only the student after training avoids the inference latency. All experiments are run on a workstation equipped with four NVIDIA RTX A5000 GPUs.

We compare the proposed framework against three baselines: (1) LLM only: the LLM is queried directly with the prompt, and its action proposal is applied to the environment without any RL student agent training. (2) RL: An RL agent is trained from scratch on the simulator. (3) Pretrained RL: an RL policy is first pretrained on the simulator and then used to guide the student RL agent. These baselines isolate the benefit of having a teacher in general. The proposed framework, in which the student is trained with the composite objective of Eq. (1) using the LLM-based teacher agent: the gpt-oss-20b.

Figure 3 shows that: (1) The vanilla RL baseline improves steadily, but it requires many interactions with the environment before reaching the low-emittance, high-reward region. This behavior reflects the low sample efficiency and high exploration cost of standard RL training. (2) The LLM-only baseline provides a useful prior for the bunch-merging task,

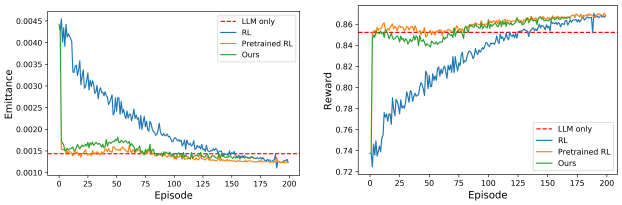

Figure 3: Training performance of the student reinforcement learning (RL) agent for AGS bunch merging. Left: evolution of the bunch emittance over training episodes. Right: corresponding evolution of the reward.

Table 1: Student RL agent's bunch-merging performance at initialization and at the best training checkpoint.

	Initial	**Best**
Emittance ($\times 10^{-3}$)	4.53	1.43
Reward	0.400	0.859

but it is not updated through environment feedback. In addition, directly using a large LLM for control can introduce significant inference latency. (3) The pretrained RL baseline performs well in the early training stage, but it requires a separate RL pretraining procedure. Thus, its strong initial performance is obtained at the cost of additional data-sampling and exploration. In contrast, the proposed framework uses the LLM teacher agent during training to guide a compact student RL agent. This framework gives the student a strong, warm start, leading to faster convergence and better early training performance than RL trained from scratch.

Table 1 summarizes the student RL agent's bunch-merging performance during training. The theoretical ground truth is the sum of the two initial bunch emittances. At initialization, the student agent produces a merged emittance of 4.53×10^{-3}, corresponding to a 355.73% deviation from the ground truth. After training, the best policy reduces the merged emittance to 1.43×10^{-3}, a 43.86% deviation from the theoretical optimum (i.e., 43.86% above the theoretical minimum), while the reward improves from 0.400 to 0.859. The LLM-guided student agent, therefore, drives the bunch merges close to the theoretical ground truth.

CONCLUSION

In this work, we proposed an active supervision framework for AGS RF bunch merging that uses an LLM-based teacher to guide the training of a compact, domain-specific student RL policy. The teacher transfers bunch-merging prior knowledge, retrieved successful merge examples, and action-space constraints. The student then refines its policy through environment feedback. On a Bmad-based AGS simulator, the framework converges faster and reaches stronger early performance than RL trained from scratch, while requiring only the lightweight student policy at deployment. Future work will validate the framework against live machine data and extend the action space to include RF phase programs.

REFERENCES

[1] W. W. Mackay *et al.*, "Commissioning and Future Plans for Polarized Protons in RHIC", in *Proc. PAC'01*, Chicago, IL, USA, Jun. 2001, pp. 24–26. doi:10.1109/PAC.2001.987421

[2] V. Schoefer *et al.*, "RHIC Polarized Proton-Proton Operation at 100 GeV in Run 15", in *Proc. IPAC'15*, Richmond, VA, USA, May 2015, pp. 2384–2386, 2015. doi:10.18429/JACoW-IPAC2015-TUPWI060

[3] A. Zelenski, G. Atoian, D. Raparia, J. Ritter, A. Kolmogorov, and V. Davydenko, "High-intensity polarized h− ion source for the RHIC SPIN physics", *AIP Conf. Proc.*, vol. 1869, no. 1, p. 030015, Aug. 2017. doi:10.1063/1.4995735

[4] A. Zelenski, G. Atoian, T. Lehn, D. Raparia, and J. Ritter, "High-intensity polarized and un-polarized sources and injector developments at BNL Linac", *AIP Conf. Proc.*, vol. 2373, no. 1, p. 070003, Jul. 2021. doi:10.1063/5.0057677

[5] H. Huang *et al.*, "Overcoming horizontal depolarizing resonances with multiple tune jumps", *Phys. Rev. ST Accel. Beams*, vol. 17, no. 8, p. 081001, Aug. 2014. doi:10.1103/PhysRevSTAB.17.081001

[6] N. Tsoupas, H. Huang, W. W. MacKay, F. Meot, T. Roser, and D. Trbojevic, "Bnl alternating gradient synchrotron with four helical magnets to minimize the losses of the polarized proton beam", *Phys. Rev. ST Accel. Beams*, vol. 16, no. 4, p. 043501, Apr. 2013. doi:10.1103/PhysRevSTAB.16.043501

[7] K. Zeno, "The 2022 polarized proton run in the injectors", BNL, Upton, NY, USA, Rep. BNL-223784-2022-TECH C-A/AP/685, Oct. 2022. doi:10.2172/1902934

[8] Aug., 1998. https://www.osti.gov/biblio/638222

[9] Y. Gao *et al.*, "Exploring reinforcement learning for optimal bunch merge in the ags", in *New York Scientific Data Summit 2025*, pp. 13–16. doi:10.1137/1.9781611978933.4

[10] Y. Gao *et al.*, "Optimization of AGS bunch merging with reinforcement learning", in *Proc. IPAC'24*, Nashville, TN, USA, May 2024, pp. 1782–1785. doi:10.18429/JACoW-IPAC2024-TUPS53

[11] V. Kain *et al.*, "Sample-efficient reinforcement learning for cern accelerator control", *Phys. Rev. Accel. Beams*, vol. 23, no. 12, p. 124801, Dec. 2020. doi:10.1103/PhysRevAccelBeams.23.124801

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning, second edition: an introduction*. MIT Press, Cambridge, MA, USA, 2018. https://books.google.com/books?id=sWV0DwAAQBAJ

[13] T. B. Brown *et al.*, "Language models are few-shot learners", *arXiv*, 2020. doi:10.48550/arXiv.2005.14165

[14] OpenAI *et al.*, "Gpt-4 technical report", *arXiv*, 2024. doi:https://doi.org/10.48550/arXiv.2303.08774

[15] J. Kaiser, A. Eichler, and A. Lauscher, "Large language models for human-machine collaborative particle accelerator tuning through natural language", *arXiv*, 2024. doi:10.48550/arXiv.2405.08888

[16] A. Sulc, R. Kammering, A. Eichler, and T. Wilksen, "Pacuna: automated fine-tuning of language models for particle accelerators", *arXiv*, 2023. doi:10.48550/arXiv.2310.19106

[17] Z. Zhou, B. Hu, C. Zhao, P. Zhang, and B. Liu, "Large language model as a policy teacher for training reinforcement learning agents", in *Proc. 33th Int. Joint Conf. Artif. Intell.*, Jeju, South Korea, 2024. doi:10.24963/ijcai.2024/627

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor", in *Int. Conf. Mach. Learn.*, Stockholm, Sweden, Jul. 2018, pp. 1861–1870.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, Cambridge, MA, USA, 2016.

[20] D. Sagan, "Bmad: a relativistic charged particle simulation library", *Nucl. Instrum. Methods Phys. Res. A*, vol. 558, no. 1, pp. 356–359, 2006. doi:10.1016/j.nima.2005.11.001

[21] OpenAI *et al.*, "Gpt-oss-120b and gpt-oss-20b model card", *arXiv*, 2025. doi:10.48550/arXiv.2508.10925

[22] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks", *arXiv*, 2021. doi:10.48550/arXiv.2005.11401