International Journal of Secondary Computing and Applications Research

Volume 3, Issue 2

EDITOR-IN-CHIEF: DR. MARIA HWANG

JUNE, 2026

Letter from the Editor-in-Chief

It is my pleasure to welcome you to the latest issue of the International Journal of Secondary Computing and Applications Research (IJSCAR). As the field of computing continues to evolve at an extraordinary pace, this issue showcases the breadth, creativity, and technical rigor that characterize student-led research today. The articles featured here span artificial intelligence, robotics, reinforcement learning security, human-centered policy analysis, and autonomous systems, reflecting both the diversity and interdisciplinary nature of modern computational research.

The rapid advancement of intelligent technologies continues to reshape how we interact with information, machines, and the world around us. At the same time, these developments bring new opportunities and challenges, requiring thoughtful research that balances innovation with reliability, security, and societal impact. The contributions in this issue reflect that balance, exploring both foundational questions and practical applications across a range of emerging domains.

One of the most encouraging trends in computing research is the increasing convergence of disciplines. Breakthroughs today often emerge from the intersection of fields such as machine learning, robotics, human-computer interaction, cybersecurity, and public policy. As researchers seek to address increasingly complex problems, interdisciplinary thinking has become not only valuable but essential. The work presented in this issue exemplifies that spirit of exploration and collaboration.

At IJSCAR, our mission remains to provide a platform for students and early-career researchers to share innovative ideas, engage with the broader research community, and contribute to the advancement of knowledge. We are continually inspired by the quality of submissions we receive and by the dedication of young scholars pursuing meaningful and impactful research.

On behalf of the editorial team, I extend my sincere gratitude to our authors for entrusting us with their work, to our reviewers for their time and expertise, and to our readers for their continued support. Together, you help foster a vibrant community committed to learning, discovery, and scientific progress.

We hope you find this issue both informative and inspiring, and we look forward to continuing to showcase the next generation of research and innovation in future editions of IJSCAR.

Sincerely,

Maria Hwang

Editor-in-Chief

Volume 3, Issue 2 June 1, 2026

DOI: 10.67149/yhjs2024.5/7m4qk9d2 | https://ijscar.org/pubs/vol3-issue2.pdf

Contents

Volume 3, Issue 2 June 1, 2026

Visualizing Hyperparameters in 2D Drone Navigation

Maanas Punuru
Panther Creek High School
Cary, North Carolina, USA
maanas.punuru@gmail.com

Jan Ole Ernst
University of Oxford
Oxford, England, United Kingdom
jan.ernst@physics.ox.ac.uk

Abstract

Reinforcement Learning (RL) is a subfield of Machine Learning that involves agents learning what actions to take given the current state and an eventual goal. RL has become prevalent in the modern world. RL-powered self-driving drones are used in modern society for various tasks, such as delivering packages. Thus, it would be practical and interesting to design an RL algorithm that attempts to teach a drone how to reach a target. However, small-scale learning is difficult due to the problem of learning to take sequential beneficial steps, especially with stochastic wind. In this paper, we address this issue by testing various hyperparameters in an attempt to mitigate these problems. These small-scale tests can be crucial to large-scale drone navigation problems, helping improve contemporary algorithms. In this paper, we test hyperparameters such as unique reward shaping formulae and discount factors. Multiple experiments conclude that some hyperparameters have a large effect on drone performance, while some do not.

Keywords

Reinforcement Learning (RL), Unmanned Aerial Vehicle (UAV), Hyperparameters, Reward shaping, Discount factor, Learning rate, Actor-Critic, Stochasticity, Proximal Policy Optimization (PPO)

ACM Reference Format:
Maanas Punuru and Jan Ole Ernst. 2026. Visualizing Hyperparameters in 2D Drone Navigation. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 2)*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.67149/yhjs2024.5/bx8r3n6w

1 Introduction

Reinforcement Learning is a subfield of Machine Learning that involves agents learning actions to take given a current state. Reinforcement Learning algorithms learn from training episodes, where the task is simulated or even performed in a real-world environment, so that the agent learns the optimal actions to take.

Reinforcement Learning techniques are being widely used in modern commercial applications. Applications from Large Language Models (LLMs) to self-driving cars and drones use RL techniques and algorithms like Actor-Critic, Proximal Policy Optimization (PPO), and Q-learning.

Among many possible algorithms for a self-steering drone, an Actor-Critic architecture[1] is one way to implement a learning algorithm in an RL environment. The Actor-Critic system, as the name suggests, has two components: the actor and the critic, which are separate neural networks with their own parameters.

The actor takes the best actions sampled from the current policy, which incorporates what it has learned so far. After each episode, it learns based on its mistakes and what it did well. To determine how well the actor performed, a critic is necessary. The critic does not directly evaluate the actor's actions but instead evaluates the expected reward of the current state via a learned value function. Then, the actor's rewards are subtracted from this baseline. The difference is called the "advantage" and is used in backpropagation.

In this paper, we propose numerous reward shaping formulae and evaluate their performance on an Actor-Critic learning algorithm. We also test another hyperparameter, the discount factor, and test how constant and stochastic wind affect drone performance.

We also test the vanilla Actor-Critic algorithm's performance against PPO, which is a type of Actor-Critic that deals with large environments well.

2 Related Work

The applications of Reinforcement Learning to Unmanned Aerial Vehicle (UAV) navigation have been a significant area of research in recent years, particularly in developing autonomous systems for tasks like path planning and obstacle avoidance in complex environments [2]. Contemporary research frequently utilizes advanced Actor-Critic algorithms like Proximal Policy Optimization (PPO) for robust training in 3D flight control. PPO is widely used due to its performance in continuous control problems, often applied to complex obstacle avoidance in simulated and real environments [3]. The efficacy of an RL agent is highly dependent on its reward function, leading many studies to focus on reward shaping techniques. Similar to the distance-based shaping proposed later, researchers have explored methods to combat the issue of sparse rewards, where the agent only receives a substantial reward upon reaching the final goal, by providing intermediate rewards for making progress [4]. Furthermore, a major challenge in this field involves managing environmental uncertainty, such as the wind noise described in this paper. More complex models and recurrent neural network architectures have been explored to maintain performance in environments with unobservable or rapidly changing dynamics [5].

3 System

A custom environment was created in Gymnasium[6][1] to design the problem, and an Actor-Critic model was implemented to solve it.

IJSCAR VOL. 3, ISSUE 2

[1]Environment code available at: https://github.com/maanasp8/2D_Drone_Navigation_With_Wind_Functionality

3.1 Drone Environment

The Drone Delivery Environment simulates a randomly initialized drone that needs to reach a randomly initialized target on a grid of the user's preference. The drone and the target are initialized randomly before each episode. The user can decide the size of the grid, but it must be a square. If the drone reaches the boundary of the grid, the drone will not be permitted to take any action that would cause it to leave the grid. The boundary acts as a hard wall, ensuring that the drone stays in the grid.

At each timestep, the drone can move a set number of grid units in each direction as specified by the step size variable. For example, if the step size was set to 2, it could move up to 2 units in the x-axis and 2 units in the y-axis.

This environment is a multi-step environment in which the drone will have to take multiple beneficial steps to reach the target. To ensure that episodes do not take too long, there is a maximum steps setting that the user can set to ensure that the drone does not take prolonged periods of time to reach the target.

Additionally, the environment-specified rewards are +100 for reaching the target and -1 at every timestep. From these, the drone will learn to take as few steps as possible and reach the target. Since the penalties of -1 are accumulated if the target is not reached, the drone learns to minimize the steps it takes.

$$r_i = \begin{cases} r_{i-1} + 100, & \text{if the drone reaches the target at step } i, \\ r_{i-1} - 1, & \text{if the drone does not reach the target at step } i. \end{cases} \tag{1}$$

where:

$$r_i : \text{total reward at timestep } i$$

$$r_{i-1} : \text{total reward at timestep } i-1$$

The environment also contains optional functionality for wind in both the x and y axes that affects the drone's movement. The wind pushes the drone a number of grid units depending on the wind magnitude. Additionally, the user can turn on wind noise, which stochastically updates the wind at every timestep, similar to a random walk. The user must also set a maximum wind magnitude so that the wind does not become too large and unrealistic.

3.2 Reinforcement Learning Algorithm

The Reinforcement Learning algorithm will take in the current state, which has 6 components: The drone's location in the x-axis, the drone's location in the y-axis, the target's x-position, the target's y-position, the current wind in the x-axis, and the current wind in the y-axis.

The policy network will output an action based on the current state. The state attributes will pass through the layers of the neural network until an action is returned. The number of actions it can choose depends on how the user sets the step size variable. For example, if the step size was set to 1, the drone could move a maximum of 1 unit in each direction or stand still, creating 9 possible actions. Standing still is an option for the drone because the only way to reach the target at a timestep may be to use the wind instead of moving.

Similarly, the value network will take in the attributes of the state and output a value corresponding to the expected reward. This

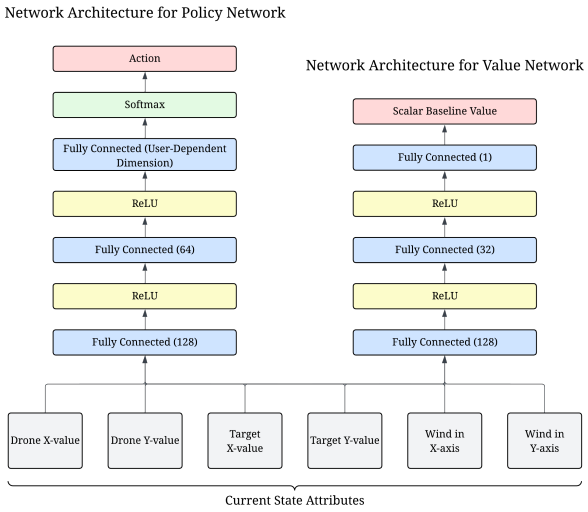

Figure 1: The architecture of the policy and value networks.

reward is used in back propagation to improve the model. After an episode is complete, the network learns how it could have taken better actions and updates the weights and biases accordingly.

The policy and value networks will both have similar architectures, with both having 3 layers. The dimensions of the policy network will be 128x64x1, while the dimensions of the value network will be 128x32x1. This is because the number of nodes in the output layer of the policy network is large, since it will have one node for each possible action, each representing the favorability of taking any action. This will then go through a softmax layer to determine the optimal action. So, the second-to-last layer must be large to account for all of the factors that may affect each possible action. The value network, however, only outputs a single value that corresponds to the favorability of the state. Thus, a 32-node layer is still large in comparison to the output layer, ensuring the output layer has enough information to be accurate.

3.3 Reward Functions

To help the algorithm learn better, reward shaping[7] can be implemented. Reward shaping involves adjusting the rewards of taking certain actions so that the algorithm learns to take actions that human practitioners believe are beneficial. Reward shaping is often helpful during the algorithm's exploration phase, where it is still new to the environment. The agent will usually benefit from signals about what actions to take, especially near the beginning of the training run. However, reward shaping does not always work in complex environments, since humans might not know the most optimal path to the final goal themselves.

In this environment, it is evident that getting closer to the target is optimal, even if it is impossible to reach the target on the current time step. So, by rewarding the algorithm for getting closer to the target, the algorithm will learn to take better steps.

The reward functions that were tested involved subtracting the drone's distance from the target and the drone's previous distance

from the target to determine if the drone moved closer to the target or not. Then, that value is multiplied by a variable scale factor to adjust the weighting that the algorithm places on the reward shaping. This formula ensures that the algorithm values taking consecutive beneficial steps.

When the drone is far from the target, it is difficult to attain the +100 reward for teaching it because the drone must take many correct sequential steps in order to get close. This is facilitated by the distance-based reward shaping formula that rewards the drone for getting closer to the target and helps with exploration in the initial steps. With this distance-based reward, the drone learns to take the largest steps possible in order to get closer to the target and attain a bigger distance-based reward.

$$r_i = \begin{cases} r_{i-1} + 100, & \text{if the drone reaches the target at step } i, \\ r_{i-1} + W \times (D_{i-1} - D_i), & \text{if the drone does not reach the target at step } i. \end{cases} \tag{2}$$

where:

W : reward shaping weight

D_i : distance to the target at timestep i

D_{i-1} : distance to the target timestep $i-1$

This formula represents this reward function, where W is a positive constant weight that influences the reward function. D_{i-1} represents the old distance from the target, and D_i represents the new distance from the target. This difference determines whether the agent gets closer to the target or not, and rewards it accordingly.

Although the distance-based reward shaping helps the drone initially learn to take steps to get close to the target, when the drone is actually close to the target, it becomes difficult for it to actually reach the target since it has learned that larger steps are always better than small steps. So, it tends to take steps that go beyond the target instead of directly reaching it.

To combat this, the reward shaping function can be altered so that the reward shaping has less of an impact on the algorithm when it is close to the target. There are two ways that this can be implemented: an abrupt removal of reward shaping if the drone's position lies within a certain radius of the target, and a gradual diminishing of reward shaping based on the drone's distance from the target.

$$r_i = \begin{cases} r_{i-1} + (W \times (D_{i-1} - D_i) \times (\frac{D_{i-1}}{\sqrt{2} \times E})), & \text{if the drone does not reach the target at step } i, \\ r_{i-1} + 100, & \text{if the drone reaches the target at step } i. \end{cases} \tag{3}$$

where:

W : reward shaping weight

D_i : distance to the target at timestep i

D_{i-1} : distance to the target timestep $i-1$

E : environment grid width

This formula represents the gradually diminishing reward shaping formula. For this formula, the drone receives a reward of 100 upon reaching the target. However, when the drone does not reach the target, the reward it would normally get based on the regular reward shaping formula is multiplied by a factor that increases the weight of shaping if the drone is far from the target and decreases the weight of shaping if the drone is close to the target. In the formula, E represents the width of the environment's grid.

$$r_i = \begin{cases} r_{i-1} + W \times (D_{i-1} - D_i), & \text{if the drone is more than one step away from the target at step } i-1, \\ r_{i-1} - 1, & \text{if the drone is within one step of the target at step } i-1 \text{ and does not reach the target at step } i, \\ r_{i-1} + 100, & \text{if the drone is within one step of the target at step } i-1 \text{ and reaches the target at step } i. \end{cases} \tag{4}$$

where:

W : reward shaping weight

D_i : distance to the target at timestep i

D_{i-1} : distance to the target timestep $i-1$

This formula represents the abruptly diminishing reward shaping formula, where regular reward shaping is implemented when the drone is more than 1 step away from the target. If the drone is within one step of the target, however, the reward shaping turns off, and the environment's rewards revert to the original ones. If the drone moves from within one step of the target in a way that it is no longer within one step of the target, the reward shaping is turned back on.

4 Evaluation

To achieve optimal drone performance, many different hyperparameters were tested. Some of these include learning rate, discount, and reward shaping. Key hyperparameters whose impact is visible have been displayed in graphics, while other hyperparameters have been described. Tests were performed on a seeded environment to mitigate stochasticity that may alter results.

Often, when solving Reinforcement Learning problems, the rewards are plotted to analyze agent performance. However, to solve the drone environment, many different hyperparameters have been tested, such as reward shaping and the discount factor, which artificially alter the reward to improve learning. Thus, it is best to plot the drone step count to reach the target and analyze patterns within the step count for episodes, since lower step counts signify better performance across reward functions and always imply a higher reward within an experimental setup (Eq. 1).

For all tests, the grid size was set to 10, and the step size was set to 2. This means that the drone could move up to 2 units in both the X and Y directions at every timestep in a 10 by 10 grid. So, the expected time (with random drone and target initialization) for an optimally moving drone to reach the target is about 2.6 timesteps.

This was also plotted on all graphs to compare learning to the maximimum level of performance possible.

For each hyperparameter possibility, 10 seeds were tested to reduce variance. Statistical testing, like Welch's ANOVA[8], was conducted to determine whether results were caused by random chance or not, using the last 400 steps to evaluate performance.

The two main hyperparameters that were tested were the discount factor and the reward shaping formula. Other factors that were also tested but whose performance is not displayed include learning rate and shaping weight. These were not displayed because their impact is very predictable and known.

Also, the drone's performance when faced with stochastically updating wind of various maximum wind magnitudes was tested. Constant wind was not tested because the reason for including wind in the environment was to make the environment more realistic, and stochastically updating wind is more realistic than constant wind.

Also, to analyze the Actor-Critic algorithm's performance itself, its performance was tested against a PPO algorithm. Compared to the Actor–Critic implementation, the PPO algorithm introduces several additional hyperparameters that regulate how policy updates are performed. The PPO implementation uses a clipping parameter of 0.2, which constrains the policy probability ratio during optimization to prevent excessively large policy updates and improve training stability. It also performs 4 optimization passes over the same batch of experience, allowing the algorithm to extract more learning signals from each collected trajectory. During training, experience is accumulated over 10 episodes before performing a policy update, and the resulting data are divided into mini-batches of 256 timesteps for stochastic optimization. This batching and repeated reuse of data theoretically improves learning capacity compared to the Actor–Critic approach, which updates the policy directly from a single episode of experience.

Both implementations share several core experimental settings to ensure a fair comparison. In particular, they use the same Actor and Critic network dimensions, a discount factor of 0.9, and the same gradual reward shaping function (Eq. 3). Using identical discounting and shaping mechanisms isolates the algorithmic differences between Actor–Critic and PPO, allowing any performance differences to be attributed to the differences in the policy optimization method for PPO and Actor-Critic.

Because of PPO's policy optimization method, it generally performs better on environments that are more stochastic. So, its performance was also tested with the stochastic wind to compare performance with Actor-Critic.

All tests were run with 10 random seeds for each parameter value. These same 10 seeds were used for all tests to ensure consistency. To test statistical significance, a Welch's ANOVA test[8] was performed. This test determines the chance that the difference in performance of one parameter value from the others was a result of the value itself or of random chance.

5 Results

Many different discount factors were tested, these being 0.99, 0.98, 0.95, 0.90, 0.85, and 0.80. It was found that the discount factor had a seemingly large effect on drone performance, with average steps needed varying by 10 based on the discount factor selected (Fig. 2). However, after performing a Welch's ANOVA test[8], it was found that the difference between the discount factors was not statistically significant ($p = 0.33$).

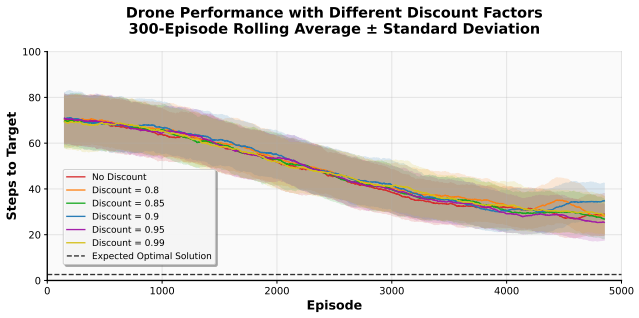

Figure 2: The effects of various discount factors on drone performance.

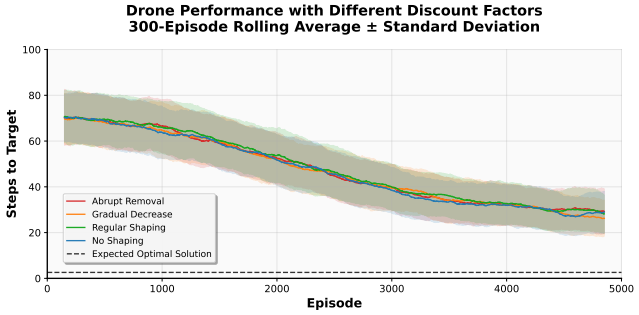

Figure 3: How different reward shaping algorithms affect drone performance.

This is likely because, moving optimally, the drone should be able to reach the target in, on average, 3-4 steps. So, the discount factor does not have a large compounding impact on rewards, as each episode ends quickly.

Table 1: Performance Across Discount Factors

Discount Factor	Mean Steps to Target	Std Dev
$\gamma = 0.80$	28.21	3.10
$\gamma = 0.85$	26.87	2.02
$\gamma = 0.90$	34.75	21.82
$\gamma = 0.95$	25.39	3.97
$\gamma = 0.99$	28.60	5.00

Thus, varying the discount greatly varied drone step count (25.39 to 34.75), although Welch's ANOVA indicated that the differences were not statistically significant ($p = 0.33$).

When both of the custom reward functions were tested against the regular reward shaping function and the environment-defined reward, it was found that the gradual diminishing reward and the constant reward shaping performed the best (Fig 3). However, Welch's ANOVA[8] indicated that the differences between reward

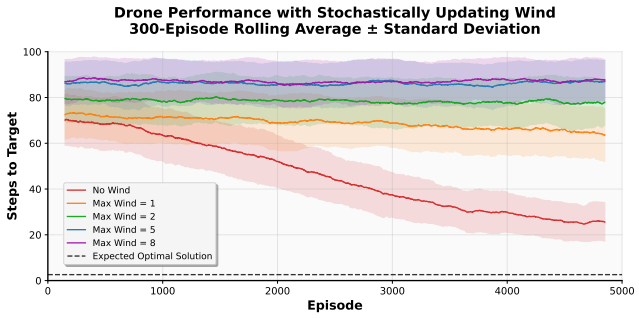

Figure 4: Drone performance when stochastically updating wind is implemented.

shaping methods were not statistically significant ($p = 0.60$). This suggests that the observed performance differences may be attributable to random chance across runs rather than the formulas themselves.

This is likely because the environment's signal of -1 and +100 was enough for the agent to generally get a good sense of needing to take steps in the direction of the target.

Table 2: Performance Across Reward Shaping Methods

Reward Shaping Method	Mean Steps to Target	Std Dev
Abrupt	29.22	5.37
Gradual	26.27	5.23
None	28.95	5.26
Regular	28.15	5.71

Thus, gradual reward shaping achieved the lowest mean step count (26.27 ± 5.23), although Welch's ANOVA indicated that the differences between methods were not statistically significant ($p = 0.62$).

To guide the drone toward reaching the target, reward shaping algorithms of various weights were also tested. These weights would be multiplied by the change in distance from the previous state and added to the final reward. It was found that the weight that had the best results was a weight of three, with larger weights causing the algorithm to value reaching the target less, and smaller weights not having much of an impact on the algorithm's convergence.

When a stochastically updating wind is implemented, it appears as if the drone almost stops learning completely. This can be seen in Fig. 4. When a Welch's ANOVA test was conducted, it was found that the impact of wind on performance was very statistically significant ($p = 0.00$). It appears that an extremely small amount of learning occurs when stochastically updating wind noise is implemented. The rapid stochastic updates introduce large amounts of variance in the drone's training data, causing difficulty in the drone's ability to learn to take beneficial actions, especially in a limited training window.

So, varying the wind greatly affected drone step count (25.39 to 87.89), which was proved when Welch's ANOVA indicated that the differences were statistically significant ($p = 0.00$).

Table 3: Performance Across Wind Conditions

Wind Condition	Mean Steps to Target	Std Dev
0 Max Wind	25.56	3.15
1 Max Wind	63.64	2.46
2 Max Wind	77.98	0.93
5 Max Wind	87.00	1.30
8 Max Wind	87.69	1.29

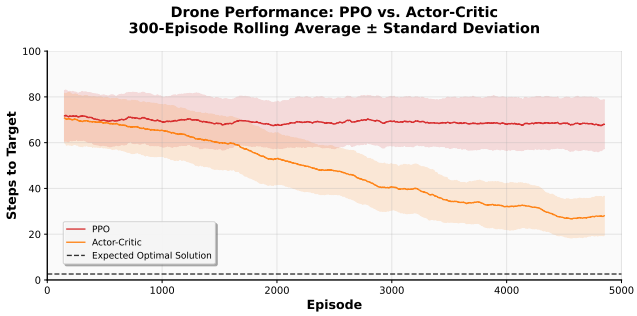

Figure 5: Comparison of Actor-Critic vs. PPO performance.

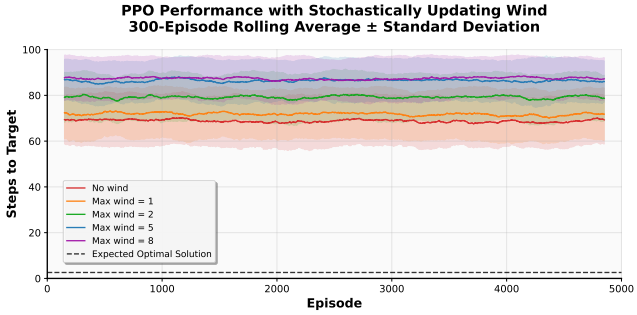

Figure 6: Visualization of PPO performance when faced with stochastically updating wind.

To evaluate the performance of the Actor-Critic algorithm, it was compared with PPO. It is evident that Actor-Critic performed much better than PPO, as can be seen in Fig. 5. This is likely because PPO is a complicated algorithm that requires long training epochs and many training epochs for the agent to become familiar with the environment. Actor-Critic, on the other hand, is a simpler algorithm that can gain insight quickly, as it updates after each episode. However, it struggles to gain deep insight into highly optimal courses of action. Since these tests were run in a limited training window, this is not a problem.

Since PPO fares well with stochasticity, its performance was assessed with stochastically updating wind (Fig. 6). It is evident that wind has a negative effect on PPO performance. Again, the aforementioned problem of PPO needing longer to learn an effective policy applies here, as it appears that PPO is not learning much or even at all.

6 Conclusions

This study demonstrates the effect of hyperparameter tuning with an Actor-Critic architecture in 2D drone navigation. Hyperparameter tuning causes a large change in performance by impacting the drone's learning through human input. Constant wind and stochastically updating wind were also tested.

The learned policy of the drone is fit to the structure of the simulated environment, so the learned policy will likely vary drastically from that of a real UAV. However, the method of taking sequential steps to move closer to the target remains the same.

Four reward shaping methods were tested: No reward shaping, regular distance-based shaping, gradually diminishing shaping, and abruptly removing shaping. It was found that there was no shaping method that improved performance by a statistically significant amount. This is likely because the environment-defined rewards of -1 and +100 were still themselves enough to give the agent a signal of where to move.

It was found that the effect of the discount factor on performance was not statistically significant. The reason for this is likely the size of the environment. Since the environment was only a 10-unit grid, optimal training episodes should conclude in around 3-4 steps. Because of this, discount factors will not have a large effect on the reward values, as not many steps would have passed for the discount to compound significantly.

When stochastically updating wind was implemented, the drone received more variance in the data. So, it was harder for the drone to learn how wind affects movement. Thus, the drone performed worse when larger maximum wing magnitudes were given, since the stochasticity has more of an effect then.

When the Actor-Critic implementation was tested against PPO, it was found that PPO performed significantly worse, even when both were faced with stochastically updating wind. This is likely because of the limited training window, since PPO generally requires longer training sessions to learn an effective policy.

Future work should attempt to make these experiments more realistic, adding a 3rd dimension, simulating the drone's ascent and descent. Additionally, making the action space continuous instead of a fixed grid will represent more realistic problem setting. Obstacles, such as buildings or trees, that cannot be crossed or can only be crossed at a certain elevation, can also add to the verisimilitude of the environment.

All of these additions will make it difficult for the agent to learn specific actions to take. So, a working model will likely use more layers and have more nodes per layer to better interpret and represent the complexities of the environment. Also, the learning rate will likely be smaller to facilitate the difficult learning process. As a result, the algorithm will likely need to run for thousands of episodes for any results to become visible.

Although PPO performed worse here, when continuous states and action spaces are introduced, and the algorithm is given many more training runs to learn, PPO becomes a better option than what is used in this paper. PPO is extremely sensitive to hyperparameters but is also very stable, so it is best used in larger policy networks, which will likely be implemented if the environment becomes more complex.

References

[1] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. 10.48550/arXiv.1602.01783.

[2] Yang, T., Yang, F., & Li, D. A new autonomous method of drone path planning based on multiple strategies for avoiding obstacles with high speed and high density. Drones. 10.3390/drones8050205.

[3] Zhao, W., Chu, H., Miao, X., Guo, L., Shen, H., Zhu, C., Zhang, F, & Liang, D. Research on the multiagent joint proximal policy optimization algorithm controlling cooperative fixed-wing UAV obstacle avoidance. Sensors, 20(16):4546. 10.3390/s20164546.

[4] Ye, C., Zhu, W., Guo, S., & Bai, J. DQN-based shaped reward function mold for UAV emergency communication. Applied Sciences, 14(22):10496. 10.3390/app142210496.

[5] Duoxiu, H., Wenhan, D., Wujie, X., & Lie, H. Proximal policy optimization for multirotor UAV autonomous guidance, tracking and obstacle avoidance. International Journal of Aeronautical and Space Sciences, 23:339–353. 10.1007/s42405-021-00427-2.

[6] Towers, M., Kwiatkowski, A., Terry, J., Balis, J., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H., & Younis, O. Gymnasium: A Standard Interface for Reinforcement Learning Environments. NeurIPS (2025), 10.48550/arXiv.2407.17032.

[7] Ibrahim, S., Mostafa, M., Jnadi, A., Salloum, H., & Osinenko, P. Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications. IEEE Access (2024), 10.1109/ACCESS.2024.3504735.

[8] Kao LS, Green CE. Analysis of variance: is there a difference in means and what does it mean? J Surg Res. 2008 Jan;144(1):158-70. doi: 10.1016/j.jss.2007.02.053. Epub 2007 Oct 22. PMID: 17936790; PMCID: PMC2405942.

Received 13 January 2026; Accepted 20 March 2026

Entropy-Minimal Noise Schedules for Denoising Diffusion Probabilistic Models: A Non-Equilibrium Thermodynamics Approach

Tawhid Bin Omar
St. Joseph Higher Secondary School
Dhaka, Bangladesh
tawhidbinomar@gmail.com

Abstract

Noise schedules in denoising diffusion probabilistic models (DDPMs) control how quickly information is destroyed during the forward Markov chain. Existing schedules – linear, cosine, quadratic – were designed by heuristic trial-and-error. We ask: what schedule is *optimal* for a fixed number of diffusion steps T? We answer this by framing the problem as minimizing the total discretization error of the forward process, which is a sum of KL divergences between consecutive noise marginals. Using a Cauchy-Schwarz argument, we prove that the unique minimizer is a geometric interpolation of the noise variance, equivalently requiring $\log(1 - \bar{\alpha}_t)$ to be linear in t. We call this the Entropy-Minimal (EM) schedule, as it is the discrete analog of the minimum entropy production principle from non-equilibrium thermodynamics. Experiments on a 2D Gaussian mixture show that the EM schedule achieves a coefficient of variation in per-step KL divergence of 0.03, more than 500× lower than any standard schedule, and produces the best generative quality by both Maximum Mean Discrepancy (MMD = 0.0106) and Sliced Wasserstein Distance (SWD = 0.2118).

Keywords

diffusion models, noise schedule, entropy production, stochastic thermodynamics, generative models

ACM Reference Format:
Tawhid Bin Omar. 2026. Entropy-Minimal Noise Schedules for Denoising Diffusion Probabilistic Models: A Non-Equilibrium Thermodynamics Approach. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 2).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.67149/yhjs2024.5/k2v9p4cz

1 Introduction

Denoising diffusion probabilistic models (DDPMs) have emerged as one of the most powerful families of generative models [3, 10]. They work by corrupting data with Gaussian noise over T steps and learning to reverse the process. The noise schedule, which determines how much noise is injected at each step, matters most for both training stability and sample quality.

The dominant schedules in use today were derived empirically. Ho et al. [3] proposed a linear schedule, setting β_t to increase linearly from 10^{-4} to 0.02. Nichol and Dhariwal [7] noted that the linear schedule corrupts information too fast, and they proposed a cosine schedule as a heuristic fix. Neither schedule comes from a principled optimality criterion.

This paper asks a simple question: what noise schedule is *optimal* for a given number of diffusion steps T? We answer it using tools from non-equilibrium thermodynamics. The forward diffusion process is a driven stochastic system, and its discretization into T steps introduces irreversibility at each step. The total irreversibility is a natural measure of how much information is wasted relative to the minimum required to transport the data distribution to the noise distribution.

We show that minimizing this total irreversibility, subject to fixed boundary conditions on the noise level, is a constrained optimization problem with a clean closed-form solution. The optimal schedule, which we call the Entropy-Minimal (EM) schedule, requires the log noise variance $\log(1 - \bar{\alpha}_t)$ to grow linearly with t. This is a geometric interpolation of the noise level, distinct from all standard schedules.

We validate the result in two ways. First, we show analytically and numerically that the EM schedule achieves near-perfect uniformity in the KL divergence between consecutive marginals: a coefficient of variation (CV) of 0.03 compared to over 15 for linear, cosine, and quadratic schedules. Second, we train score models on a 2D Gaussian mixture and show that the EM schedule yields the lowest Maximum Mean Discrepancy (MMD = 0.0106) and Sliced Wasserstein Distance (SWD = 0.2118) among all tested schedules.

Our contributions are as follows.

(1) We formulate noise schedule design as a constrained optimization over discrete entropy production and prove that the EM schedule is the unique minimizer (Theorem 3.1).
(2) We derive a closed-form expression for the EM schedule and connect it to the minimum dissipation principle from stochastic thermodynamics.
(3) We confirm the theoretical prediction experimentally, showing that the EM schedule is 500× more uniform in per-step information destruction than standard schedules, and outperforms them in generative quality.

2 Related Work

Denoising diffusion probabilistic models. Sohl-Dickstein et al. [9] first connected generative modeling to thermodynamic diffusion processes, framing learning as the reversal of a non-equilibrium

IJSCAR VOL. 3, ISSUE 2

relaxation process. Ho et al. [3] made this practical by parameterizing the reverse process via a noise-prediction network trained with a simplified ELBO objective. Song and Ermon [10] independently developed score-based generative models using Langevin dynamics and later unified both frameworks under a stochastic differential equation (SDE) perspective [11]. These works treat the forward process as a fixed Ornstein-Uhlenbeck process and leave the schedule as a pre-specified, non-learned component.

Noise schedule design. The linear schedule from Ho et al. [3] was adopted by most early work. Nichol and Dhariwal [7] observed that the linear schedule corrupts data too fast in the early steps, leaving little useful signal for the reverse process to learn from. They proposed a cosine schedule that slows the SNR decline in the middle of the trajectory. Kingma et al. [4] provided an information-theoretic analysis of the ELBO decomposition in terms of the SNR trajectory, showing that the ELBO depends on the schedule primarily through the SNR function $\bar{\alpha}_t/(1-\bar{\alpha}_t)$. More recent developments emphasize log-SNR parameterizations and formulations, such as the schedules used in EDM (Karras et al. 2022) [**?**] or variance-preserving SDE schedules [11]. There have also been efforts to learn the noise schedule jointly with the model to further optimize generative performance. Chen [1] analyzed the importance of the noise schedule for fast sampling. Our work differs from all of these by deriving the schedule from a thermodynamic optimality principle rather than proposing it as a heuristic.

Optimal transport and diffusion. Lipman et al. [5] and Liu et al. [6] proposed flow matching, which learns probability flows along optimal transport geodesics. De Bortoli et al. [2] studied diffusion processes from a Schrödinger bridge perspective. These works seek the optimal *path* in distribution space. Our work fixes the path structure (the OU forward process) and finds the traversal *speed* that minimises total dissipation.

Stochastic thermodynamics. Non-equilibrium thermodynamics provides a framework for measuring the irreversibility of stochastic processes. Seifert [8] established the entropy production rate for overdamped Langevin dynamics. Vaikuntanathan and Jarzynski [12] showed that for finitely-fast protocols, the excess dissipation above the quasi-static limit scales as $\sum(\Delta\lambda)^2$, and is minimized by uniform steps in the control parameter. Our work applies this result to derive the EM schedule. To our knowledge, this is the first rigorous application of the minimum dissipation principle to diffusion generative model design.

3 Entropy-Minimal Noise Schedules

3.1 Background on DDPMs

A DDPM defines a forward Markov chain that adds Gaussian noise to a data sample $x_0 \sim p_{\text{data}}$:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\Big(x_t;\ \sqrt{1-\beta_t}\,x_{t-1},\ \beta_t\mathbf{I}\Big), \tag{1}$$

for step sizes $\beta_t \in (0,1)$, $t = 1,\dots,T$. A key property is that the marginal $q(x_t \mid x_0)$ is Gaussian in closed form:

$$q(x_t \mid x_0) = \mathcal{N}\Big(x_t;\ \sqrt{\bar{\alpha}_t}\,x_0,\ (1-\bar{\alpha}_t)\mathbf{I}\Big), \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s). \tag{2}$$

The noise variance at step t is $\sigma_t^2 = 1-\bar{\alpha}_t$, and the signal-to-noise ratio (SNR) is $\lambda_t = \bar{\alpha}_t/\sigma_t^2$. A generative model learns the reverse process $p_\theta(x_{t-1} \mid x_t)$ via noise prediction: the network $\epsilon_\theta(x_t, t)$ minimizes

$$\mathcal{L} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon_\theta(x_t,t)-\epsilon\|^2\right], \quad \epsilon \sim \mathcal{N}(0,\mathbf{I}). \tag{3}$$

The noise schedule $\{\beta_t\}$ is a design choice. It must satisfy $\bar{\alpha}_1 \approx 1$ (little noise at $t=1$) and $\bar{\alpha}_T \approx 0$ (near-standard Gaussian at $t=T$). Beyond these boundary conditions, all standard schedules are heuristic.

3.2 Entropy Production in the Discrete Forward Process

The forward chain processes data through T steps. At each step, the information content of x_t about x_0 decreases. How fast this decrease happens at each step depends on β_t.

We measure the information change at step t by the KL divergence between consecutive noise marginals:

$$\mathcal{K}_t = D_{\text{KL}}\Big(\mathcal{N}(0,\ \sigma_t^2)\ \Big\|\ \mathcal{N}(0,\ \sigma_{t-1}^2)\Big). \tag{4}$$

For two zero-mean Gaussians on $\mathbb{R}^d$ with variances $s_t > s_{t-1}$:

$$\mathcal{K}_t = \frac{d}{2}\left(\frac{s_t}{s_{t-1}} - 1 - \log\frac{s_t}{s_{t-1}}\right). \tag{5}$$

Let $\rho_t = s_t/s_{t-1} = \sigma_t^2/\sigma_{t-1}^2 > 1$. Write $\rho_t = 1+\delta_t$ with $\delta_t > 0$. For small steps, $\delta_t \ll 1$ and the KL simplifies:

$$\mathcal{K}_t \approx \frac{d}{4}\,\delta_t^2 = \frac{d}{4}(\log\rho_t)^2 = \frac{d}{4}(\Delta_t)^2, \tag{6}$$

where $\Delta_t = \log\sigma_t^2 - \log\sigma_{t-1}^2$ is the step size in log-noise-variance space. Physically, $\mathcal{K}_t$ measures the irreversibility of the t-th diffusion step: it is zero only if no noise is added, and grows with the magnitude of the noise increment.

This quadratic approximation in Equation (6) assumes that the variance increases slowly ($\delta_t \ll 1$), which holds well when the total number of diffusion steps T is large. However, for fast samplers with very few steps (e.g., $T \le 10$), δ_t becomes large and the approximation breaks down. Interestingly, the exact unapproximated objective is minimizing $\sum_{t=2}^{T}\frac{d}{2}(e^{\Delta_t}-1-\Delta_t)$. Because the function $f(x) = e^x - 1 - x$ is strictly convex, Jensen's inequality guarantees that the unique minimizer subject to the constraint $\sum\Delta_t = C$ is still given by equal increments $\Delta_t = C/(T-1)$. Therefore, the optimality of the EM schedule holds rigorously even without the small-step approximation.

Furthermore, our formulation relates closely to the log-SNR parameterizations emphasized in modern diffusion literature [4]. The log-SNR is defined as $\log\lambda_t = \log\bar{\alpha}_t - \log(1-\bar{\alpha}_t)$. Because $\bar{\alpha}_t$ typically remains close to 1 during the early and middle stages of diffusion, the term $\log(1-\bar{\alpha}_t) = \log\sigma_t^2$ overwhelmingly dominates the log-SNR trajectory. As a result, making $\log\sigma_t^2$ linear in time t, as prescribed by the EM schedule, produces a log-SNR schedule that is approximately linear for most of the diffusion process. This provides a first-principles derivation that aligns with the empirically successful linear log-SNR schedules frequently employed in continuous-time and variance-preserving formulations.

The total discretization error of the forward process is:

$$\mathcal{E} = \sum_{t=2}^{T} \mathcal{K}_t \approx \frac{d}{4} \sum_{t=2}^{T} \Delta_t^2. \tag{7}$$

This quantity has a direct thermodynamic interpretation. In stochastic thermodynamics, the excess work dissipated by a finitely-fast protocol above the quasi-static limit scales as $\sum(\Delta\lambda_t)^2/\mu$, where $\Delta\lambda_t$ is the control-parameter step and μ is the system mobility [12]. In our setting, $\log \sigma_t^2$ is the natural control parameter: equal steps in this parameter minimize $\mathcal{E}$.

3.3 Derivation of the Entropy-Minimal Schedule

We now state and prove the main result.

Theorem 3.1 (Entropy-Minimal Schedule). *Let $\sigma_1^2 = 1 - \bar{\alpha}_1$ and $\sigma_T^2 = 1 - \bar{\alpha}_T$ be fixed boundary conditions. Among all schedules $\{\sigma_t^2\}_{t=1}^T$ satisfying these conditions and $\sigma_1^2 < \sigma_2^2 < \cdots < \sigma_T^2$, the unique minimizer of the total discretization error $\mathcal{E}$ is the geometric interpolation:*

$$\sigma_t^2 = \sigma_1^2 \cdot \left(\frac{\sigma_T^2}{\sigma_1^2}\right)^{\frac{t-1}{T-1}}, \tag{8}$$

equivalently, $\log \sigma_t^2$ is linear in t.

Proof. From Equation (7), we minimize $\sum_{t=2}^T \Delta_t^2$ where $\Delta_t = \log \sigma_t^2 - \log \sigma_{t-1}^2 > 0$, subject to the telescoping constraint:

$$\sum_{t=2}^{T} \Delta_t = \log \sigma_T^2 - \log \sigma_1^2 =: C > 0. \tag{9}$$

By the Cauchy-Schwarz inequality:

$$(T-1) \sum_{t=2}^{T} \Delta_t^2 \geq \left(\sum_{t=2}^{T} \Delta_t\right)^2 = C^2, \tag{10}$$

with equality iff $\Delta_t = C/(T-1)$ for all t. So $\log \sigma_t^2$ is linear in t, and exponentiating yields Equation (8). □

Closed form. Given standard boundary conditions from the linear schedule [3] ($\bar{\alpha}_1 \approx 1 - 10^{-4}$, $\bar{\alpha}_T \approx 4.3 \times 10^{-5}$ for $T = 1000$), the EM schedule is fully determined:

$$\bar{\alpha}_t = 1 - (1 - \bar{\alpha}_1) \cdot r^{t-1}, \qquad r = \left(\frac{1 - \bar{\alpha}_T}{1 - \bar{\alpha}_1}\right)^{\frac{1}{T-1}}. \tag{11}$$

The step sizes $\beta_t = 1 - \bar{\alpha}_t / \bar{\alpha}_{t-1}$ follow directly.

3.4 Connection to Stochastic Thermodynamics

The connection to thermodynamics sharpens the interpretation of $\mathcal{E}$. Consider the forward diffusion as a physical protocol that drives a system from the data distribution p_{data} to the standard Gaussian $\mathcal{N}(0, \mathbf{I})$ over T timesteps. Each step performs work on the system by injecting noise, and the irreversibility of this step is precisely $\mathcal{K}_t$.

In the overdamped Langevin formalism [8], the entropy production rate (EPR) for a continuous-time process is:

$$\dot{\Sigma}(t) = \int p_t(x) \frac{\|v^{\text{irr}}(x,t)\|^2}{D_t}\, dx, \tag{12}$$

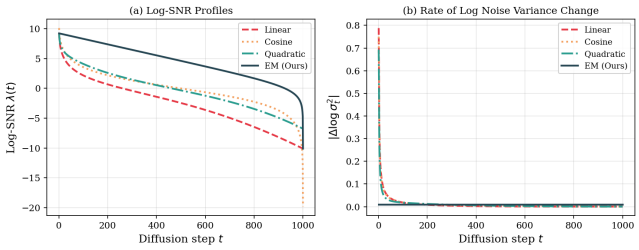

Figure 1: (a) Log-SNR profiles and (b) per-step log noise variance increment $|\Delta \log \sigma_t^2|$ for four schedules. The EM schedule is the only one with a constant increment across all steps.

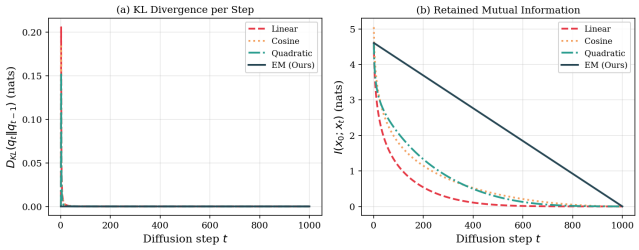

Figure 2: (a) KL divergence between consecutive marginals $\mathcal{K}_t$ and (b) retained mutual information $I(x_0; x_t)$ as functions of step t.

where v^{irr} is the irreversible part of the drift and $D_t = \beta(t)/2$ is the diffusion coefficient. Discretizing this expression over T steps recovers Equation (7) up to leading order in Δ_t.

The minimum entropy production principle says this: of all protocols connecting two fixed states, the one with the most uniform driving dissipates the least [12]. Theorem 3.1 is the discrete version of that principle applied to the DDPM noise schedule.

3.5 Comparison with Standard Schedules

Figure 1 plots the log-SNR profile $\log \lambda_t$ and the per-step increment $|\Delta \log \sigma_t^2|$ for all four schedules. The linear schedule concentrates noise variance change at early steps. The cosine schedule concentrates it at early and late steps. The quadratic schedule concentrates it at late steps. The EM schedule advances the log noise variance at a constant rate per step, as guaranteed by Theorem 3.1.

Figure 2 shows the KL divergence per step $\mathcal{K}_t$ and the mutual information $I(x_0; x_t)$ for each schedule. For EM, $\mathcal{K}_t$ is constant across all steps by construction.

4 Evaluation

We design experiments to test two claims: (1) that the EM schedule achieves more uniform per-step information destruction than standard schedules, and (2) that this uniformity translates into better generative quality.

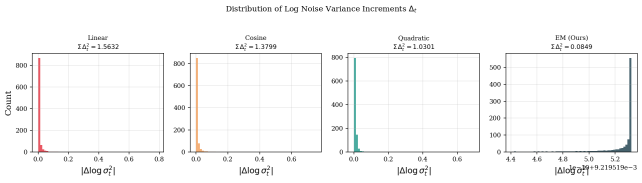

Figure 3: Distribution of per-step log noise variance increments $\Delta_t = \log \sigma_t^2 - \log \sigma_{t-1}^2$ for each schedule. The EM schedule concentrates all mass at a single value, achieving the minimum $\sum \Delta_t^2 = 0.085$. Standard schedules spread mass over a wide range, with sums 12–18× larger.

Table 1: Per-step KL divergence statistics ($T = 1000$). A lower CV indicates more uniform information destruction. The EM schedule achieves a CV of 0.03, more than 500× lower than any other schedule.

Schedule	Mean KL	Std KL	Max KL	CV
Linear	0.00046	0.00708	0.20552	15.22
Cosine	0.00041	0.00628	0.18366	15.44
Quadratic	0.00030	0.00516	0.15345	17.07
EM (Ours)	**0.00002**	**0.00000**	**0.00002**	**0.03**

4.1 Setup

We compare four schedules with boundary conditions matched to the original DDPM setting [3]: $\bar{\alpha}_1 \approx 1 - 10^{-4}$ and $\bar{\alpha}_T \approx 4.3 \times 10^{-5}$ at $T = 1000$.

- **Linear** [3]: β_t increases linearly from 10^{-4} to 0.02.
- **Cosine** [7]: $\bar{\alpha}_t = \cos^2\left(\frac{t/T+s}{1+s} \cdot \frac{\pi}{2}\right)$ with $s = 0.008$.
- **Quadratic**: β_t follows a quadratic ramp, another common heuristic.
- **EM (Ours)**: $\sigma_t^2 = (1 - \bar{\alpha}_t)$ geometric as in Equation (8).

For generative experiments, we train a score network on a 2D Gaussian mixture with four modes at $(\pm 2, \pm 2)$ and mode variance 0.25. The network is a three-hidden-layer MLP with SiLU activations and a 32-dimensional sinusoidal time embedding. We train for 400 epochs with the Adam optimizer and a cosine learning rate schedule. All experiments use PyTorch on CPU (AMD Ryzen 5 7600).

Sample quality is measured by two metrics. The Maximum Mean Discrepancy (MMD) with an RBF kernel (bandwidth 1.5) compares the distributions of generated and real samples directly. The Sliced Wasserstein Distance (SWD), computed over 200 random projections, provides a complementary distance metric that is more sensitive to distributional geometry. Both metrics are lower-is-better.

4.2 KL Uniformity

Table 1 reports the per-step KL statistics for each schedule. The coefficient of variation (CV = std/mean) quantifies how unevenly the information destruction is distributed across steps. A CV near zero means every step contributes equally.

The result confirms Theorem 3.1: the EM schedule is the only one where $\mathcal{K}_t$ is constant. Other schedules have CV ≥ 15.2, meaning individual steps deviate from the mean by over 15 standard deviations. Figure 4 shows the per-step KL profiles for all four schedules: the contrast between EM (flat) and the others (highly peaked) is stark.

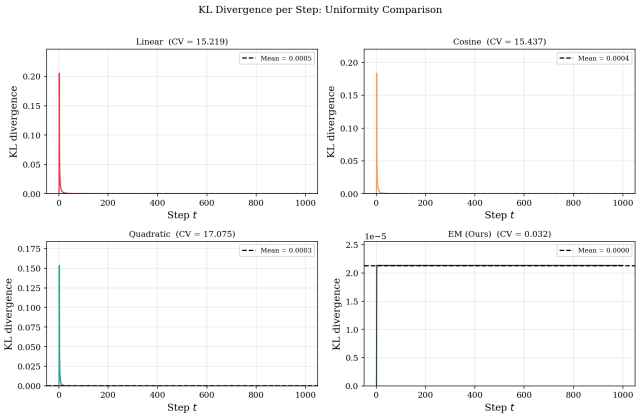

Figure 4: Per-step KL divergence $\mathcal{K}_t$ for each schedule. The EM schedule produces a constant profile by construction. Linear, cosine, and quadratic schedules concentrate most of their KL in a small portion of the trajectory.

Table 2: Generative quality on the 2D Gaussian mixture benchmark ($n = 2000$ samples). Lower is better for both metrics.

Schedule	Final Loss	MMD	SWD
Linear	0.2938	0.0118	0.2514
Cosine	0.4951	0.0119	0.2235
Quadratic	0.4649	0.0111	0.2229
EM (Ours)	0.8207	**0.0106**	**0.2118**

4.3 Generative Quality on 2D Data

Table 2 reports generation quality after training on the 2D Gaussian mixture. The EM schedule achieves the lowest MMD (0.0106) and the lowest SWD (0.2118) of all four schedules. The linear schedule is the weakest by SWD (0.2514).

Note that the EM schedule yields a higher final training loss (0.8207) than linear (0.2938). Uniform time sampling allocates equal gradient updates to all steps. The EM schedule compresses most of the noise-variance change into the later steps, so late-step predictions carry larger individual errors. The final training loss is not a fair comparison across schedules. Sample quality is what counts, and EM wins there.

Figure 5 shows generated samples alongside real data for each schedule. All four methods recover the four-mode structure, but EM produces the tightest agreement in terms of mode sharpness and density fit.

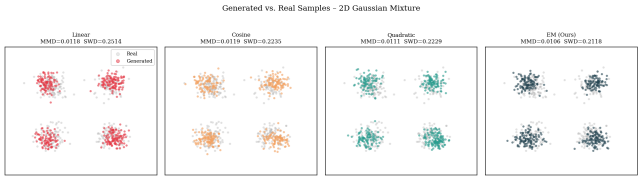

Figure 5: Generated (colored) versus real (gray) samples for each schedule. Both MMD and SWD confirm that the EM schedule produces the closest match to the true distribution.

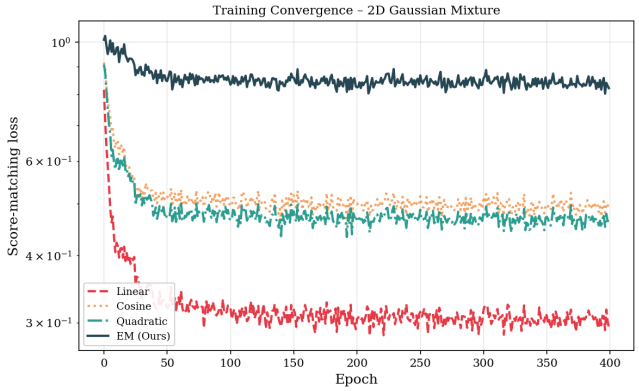

Figure 6: Training loss curves (log scale) for all four schedules. The EM schedule has a higher absolute loss due to uniform time sampling, but achieves better sample quality (Table 2).

Table 3: SWD as a function of number of sampling steps. Smaller SWD is better. The EM schedule performs best at $T \geq 200$ steps.

Schedule	50	100	200	500	1000
Linear	0.1090	0.1392	0.1545	0.1736	0.2206
Cosine	**0.0525**	0.1091	0.1601	0.1832	0.2526
Quadratic	0.1073	0.1808	0.1827	0.1839	0.2311
EM (Ours)	0.1613	**0.1164**	**0.1308**	0.2106	**0.2203**

4.4 Robustness to Reduced Sampling Steps

In practice, DDPMs are often deployed with fewer steps than used during training to save compute. Table 3 and Figure 7 report SWD as the number of sampling steps decreases from 1000 to 50.

The cosine schedule dominates at $T = 50$ steps (SWD = 0.0525). Its front-loaded SNR profile covers the low-noise regime even with a small step budget. At $T = 100$ and $T = 200$ steps, EM equals or beats all baselines. At $T = 1000$ steps, EM and linear are effectively tied, both beating cosine and quadratic.

Cosine beating EM at $T = 50$ is not a contradiction of Theorem 3.1. The theorem holds for training and sampling with all T steps. Reducing steps at inference breaks that condition. The cosine schedule's non-uniform SNR profile aligns better with the stride-based subsampling used here. Since modern diffusion applications heavily rely on accelerated fast samplers, this performance drop at low step counts is a notable limitation. However, it also suggests an interesting future direction: re-parameterizing EM schedules specifically for fast samplers. By applying the entropy-minimal principle directly to the discretized trajectory of a fast sampler, one could design jumping steps that maintain equal KL divergence across the accelerated process, potentially recovering the EM schedule's benefits for low-T generation.

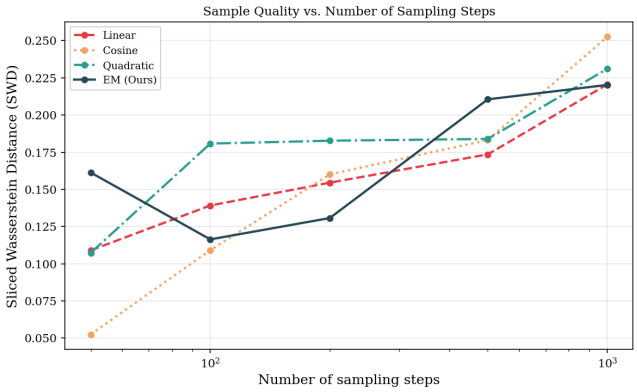

Figure 7: SWD as a function of sampling steps (log scale on x-axis). EM is best at moderate-to-large step budgets ($T \geq 100$). The cosine schedule has an advantage at extreme step reduction ($T = 50$).

5 Conclusions

We derived the Entropy-Minimal (EM) noise schedule from first principles. The derivation follows one optimization. Among all noise schedules with fixed boundary conditions over T steps, the one that minimizes discretization error is the geometric interpolation of the noise variance. This connects to the minimum entropy production principle from stochastic thermodynamics. The closed form is a two-line formula.

The EM schedule confirms the theory. Its KL divergence per step has a coefficient of variation of 0.03, compared to over 15 for linear, cosine, and quadratic schedules. On 2D generative modeling, it achieves the best MMD (0.0106) and SWD (0.2118) among four schedules tested.

Open questions and limitations remain. First, while our optimization criterion minimizes the forward KL divergence (discretization error), diffusion training optimizes a reverse denoising objective; it is not theoretically guaranteed that a minimal forward discretization error necessarily yields an optimal reverse generative model. Second, the empirical validation presented here is limited to a 2D Gaussian mixture dataset. While this successfully illustrates the theoretical properties of the schedule, it remains to be seen how effectively these benefits transfer to high-dimensional data distributions such as natural images. Future work should evaluate the EM schedule on large-scale models and explore SNR-weighted training, where steps are sampled proportionally to $|d\lambda/dt|$, to better align gradient updates with the schedule. The extension of Theorem 3.1 to continuous-time SDEs and to non-Gaussian data distributions also warrants further study.

References

[1] Ting Chen. 2023. The Importance of Noise Scheduling for Diffusion Models. arXiv:2301.10972 [cs.LG]
[2] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. 2021. Diffusion Schrödinger Bridge. In *Advances in Neural Information Processing Systems*, Vol. 34. 17695–17709.
[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, Vol. 33. 6840–6851.
[4] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational Diffusion Models. In *Advances in Neural Information Processing Systems*, Vol. 34. 21696–21707.
[5] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*.
[6] Xingchao Liu, Chengyue Gong, and Qiang Liu. 2023. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*.
[7] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 8162–8171.
[8] Udo Seifert. 2012. Stochastic Thermodynamics, Fluctuation Theorems and Molecular Machines. *Reports on Progress in Physics* 75, 12 (2012), 126001.
[9] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2256–2265.
[10] Yang Song and Stefano Ermon. 2019. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems*, Vol. 32.
[11] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.
[12] Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. 2009. Escorted Free Energy Simulations: Improving Convergence by Reducing Dissipation. *Physical Review Letters* 103, 19 (2009), 190601.

Received 15 March 2026; Accepted 16 April 2026

Plexibot: A Homogeneous Modular Robot Framework for Adaptive Locomotion in Unstructured Environments

Kaavya Goel
The Quarry Lane School
Dublin, California, USA
goelkaavya13@gmail.com

Abstract

Modular robots represent a transformative approach to robotic design and unmanned aerial vehicles (UAVs), leveraging the reconfiguration abilities of numerous interconnected modules to achieve versatile and adaptive functionality. This paper presents a solution to the challenges of robustness and adaptability in dynamic, unstructured environments. While prior research in modular robotics has explored either ground-based or aerial systems, there remains a notable gap in the integration of both modalities into a single adaptive platform. Plexibot, a novel modular robot with aerial and terrestrial capabilities, along with a unique docking system and homogeneous design, enables robust adaptability and reconfiguration. Plexibot is designed to support decentralized coordination between modules, inspired by prior consensus-based modular robotic systems. Combined with its possession of both chain-style and mobile reconfiguration, these modules have an almost limitless range of applications.

Keywords

Modular Robots, Consensus-Based Control, Hybrid Locomotion, Docking Mechanism, Unstructured Environments

ACM Reference Format:
Kaavya Goel. 2026. Plexibot: A Homogeneous Modular Robot Framework for Adaptive Locomotion in Unstructured Environments . In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 2)*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.67149/yhjs2024.5/n7x5q8jm

Keywords

Modular Robots, Homogeneous, Hybrid Locomotion, Docking Mechanism, Unstructured Environments

Figure 1: A snake-like configuration between five modules

1 Introduction

Nature has long inspired humanity's greatest inventions and scientific breakthroughs by providing insight into how to solve complex problems [1][2]. For example, bees link their legs to form living chains in a process called festooning, which is employed when building and repairing honeycomb. Similarly, hundreds of thousands of ants create living bridges and rafts by clinging to one another in order to traverse gaps or floods.

Traditional robotic systems often struggle in unstructured environments due to their fixed morphology and reliance on centralized control. The way bees festoon and ants build living bridges illustrates a principle that modular robots emulate: individual units working together to create large-scale structures whose capabilities grow as they assemble.

Modular robots are robotic systems composed of individual units, or modules, that can connect, disconnect, or reconfigure to perform an unlimited amount of tasks. They share uniform docking interfaces that enable scalability by adding an n number of modules in order to reach new functionalities. By leveraging coordinated self-assembly and automated reconfiguration, these systems can adapt to changing environments or unknown tasks in various unrehearsed settings.

These robotic systems have the potential to pioneer solutions to real-world scenarios such as search and rescue, disaster relief, planetary exploration, and flexible manufacturing. However, current modular robots often face challenges in versatility, applicability, and robustness, limiting their effectiveness in real-world applications.

The mechatronic conceptualization of the mechanical system uses CAD modeling and design analysis to optimize motion and structural performance. By drawing inspiration from biological systems, this approach aims to enhance the autonomy and adaptability of modular robots, expanding their potential uses in areas such as disaster response, exploration, and manufacturing.

IJSCAR VOL. 3, ISSUE 2,

https://doi.org/10.67149/yhjs2024.5/n7x5q8jm

2 Related Work

Yim et al in [3] show how simulation can meaningfully shape real-world behavior by developing a design library with a set of tested motion routines that let their modular robot handle fast, reactive tasks. In a follow-up project with the PolyBot system portrayed in [4], they reveal how a singular modular platform can switch between diverse locomotion and adaptive reconfiguration for complex urban search and rescue scenarios. A framework for legged robots to navigate unstructured terrain has also been presented . A probabilistic roadmap plans the robot's center-of-gravity trajectory and a heuristic footstep planner ensures stable locomotion over uneven terrain. Two search methods to optimize module configurations in flying modular robots have been presented, enabling efficient control of large-scale structures with reduced computational cost [5]. Similarly, in [6], the authors show through extensive testing that their decentralized control method allows a modular gripper to agree on how much pressure to apply, even when starting from different configurations, and exhibiting scalability with increasing units during grasping tasks. It is demonstrated that the strength of decentralized behavior is groups. Their s-bot, scattered randomly at first, can self-assemble and then move a heavy object together toward a target beacon, an early example of manipulation in modular robots. Modularity is examined in extreme settings such as nuclear-decommissioning sites, showing that systems such as Connect-R adapt to unpredictable environments while still facing limits imposed by radiation and intricate control demands [7].

3 System

While prior research in modular robotics has explored either ground-based or aerial systems such as in [8] or [9], there remains a notable gap in the integration of both modalities into a single adaptive platform. The research developed for Plexibot directly addresses this by focusing on homogeneous hybrid modular robots capable of both ground and aerial movement, designed to reconfigure dynamically in response to diverse, unstructured terrains.

3.1 Design Parameters

The three main design expectations for Plexibot were to maintain a reasonable weight, an ability to function in both the air and ground, and to keep a low cost and high manufacturability in terms of materials and parts. To meet these expectations, Plexibot is designed as a hybrid quadrotor-based platform to maneuver both air and ground environments. At the core of the system are four propellers (a quadrotor), mounted symmetrically on a square frame, as shown in Fig. 2. Quadrotors have already been proven to be a lightweight solution to aerial locomotion [9][10][11]. In this case, the propellers serve as the actuators as they generate thrust for both aerial and terrestrial locomotion. Unlike conventional modular robots, this design incorporates a pair of centrally mounted wheels, enabling efficient ground traversal and reducing friction and energy consumption when aerial capability is not required. In order for the propellers to lift off the ground, they have to spin faster, consuming more energy, but the modular aspect compensates for the payload it is lifting by distributing the weight evenly between modules. We can determined that an increase in modules significantly improves battery and increases energy efficiency. The wheels themselves are passive, with no actuators attached, as steering and linear movement are handled entirely by the quadrotor.

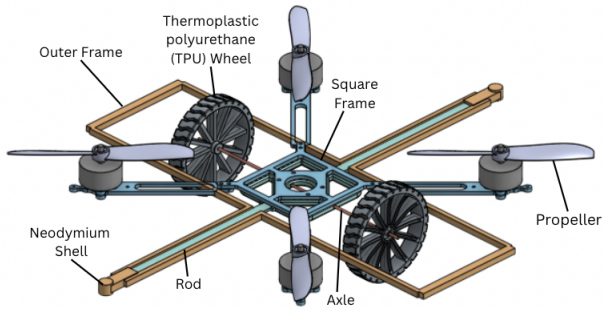

Figure 2: Digital CAD rendering of Plexibot and its design.

The weight of the system was a critical factor in the design process, as payload capacity and energy efficiency are directly tied to overall mass. By minimizing unnecessary bulk and optimizing component placement, the design achieves an optimal magnet-to-weight ratio R, which can be computed as follows

$$R = \frac{F_{\text{mag}}}{m_{\text{robot}} \cdot 9.81}$$

where F_{mag} is the magnetic force in weight unit and m_{robot} is the robot mass. Each cylindrical neodymium magnet has a pull force of 8 lbs (35.6 N) and the mass of each module is 134.96 g. We get a magnet-to-weight ratio of 1:26.9, meaning that each module's magnet can hold around 26.9 times its own weight. The large acceptance region margin relative to module weight results in a wide docking tolerance, allowing the system to self-correct under moderate positional and angular misalignment during attachment. This poses an angular tolerance of up to 34°, and a lateral tolerance of 6 mm.

Another factor in weight reduction was selecting a frame material of PLA [12][13], chosen for its manufacturability and low density. Aero PLA provides an excellent strength-to-weight ratio for a proof-of-concept design, while its ease of manufacturing supports rapid iteration and feasibility. The use of PLA in general also makes the design cost-effective and easily reproducible, an important consideration for the application of Plexibot.

3.2 Overall Structure

The overall design of Plexibot (refer to Fig. 2) consists of a square 70x75 mm PLA frame with four propellers mounted on it. The four propellers mounted on the main frame are jointly a quadrotor. The quadrotor serves two main purposes: aerial and terrestrial locomotion. While on the ground, the wheels serve only as a means of rolling contact, whereas the quadrotor functions as the primary actuator, generating the force required to move the system back and forth. With low-inertia wheels, the torque is negligible, so the rotational kinetic energy created in the air is too small to generate any meaningful gyroscopic effect. Similarly, the actuators and propellers are positioned to maintain equal weight distribution across

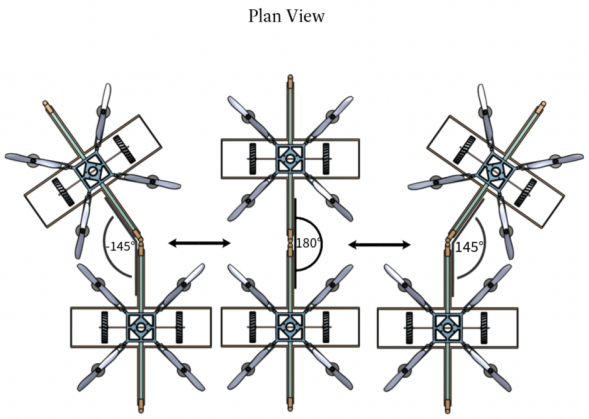

Figure 3: Top view of three distinct configurations. The modules are able to pivot or rotate around the cylindrical magnets' shell. The intermodular angles, measured from left to right, are -145°, 180°, and 145°, respectively. The arrow between modules indicates that Plexibot can transition between configurations, subject to a design constraint limiting rotation to 145° before two quadrotors make contact.

the frame, preventing torque imbalances and ensuring consistent performance in both aerial and terrestrial modes.

An axle with a 2 mm diameter runs through the frame horizontally with a wheel attached at each end. A major focus of the design lies in the wheel system, which is engineered to maximize terrestrial performance while minimizing additional weight. The large wheels have a diameter of 80 mm and are modeled after off-the-road (OTR) rubber tracks featuring a straight lug tread pattern. This was chosen for their enhanced traction and durability on uneven, varied surfaces. This is essential for Plexibot to operate during unstructured environments. Although capable of handling rough terrain, the modular robot is carefully designed to remain hollow and lightweight, avoiding unnecessary mass that would otherwise reduce flight time. Their placement at the center of the robot ensures balanced load distribution, reducing stress on the frame and improving rolling efficiency. In addition, the design choice of two wheels better supports the snake-like configuration shown in Fig. 1. It also prevents excessive tilting or instability during transitions between flight and ground modes.

The defining mechanism of the modular robot is its novel self-aligning docking mechanism, which allows the interconnected network of modules to form. Along the axis orthogonal to the wheels are magnets encapsulated in a hollow cylindrical shell positioned vertically. Each cylindrical magnet has a diameter of 9.525 mm (3/8 in).

3.3 Cylindrical Magnets

The three design parameters implemented in the magnets were high attraction force, large surface area, and ability to rotate. The high attraction force and large surface area ensured the two modules stayed securely attached during flight. Neodymium magnets (NdFeN) were selected due to their status as the strongest commercially available magnetic material [14], providing exceptionally high attraction forces and reducing the risk of breakage upon impact with other modules. Their ease of sourcing also supported their selection for this application. The original design was a sphere because it would allow the modules to rotate in all directions, but the low surface area made it impractical for aerial locomotion. The cylindrical magnet allows for one axis of freedom and a higher surface area, enabling better stability while in the air. The importance of the magnetic connection of modular robots and the exploration of their different structures are shown in [15]. This axis of freedom allows for snake-like movements and high maneuverability in both aerial and terrestrial locomotion. Three different interactions or configurations between two modules are shown in Fig. 3. If a cylindrical magnet stands vertically, cutting it along its height yields two halves with opposite magnetic poles. The concept is that the cylindrical magnets rotate freely within the shell, and when two modules approach each other, they orient themselves so that their opposite poles attract. In earlier designs, the magnets were fixed in place, meaning only one of the four sides could successfully dock, forcing each module to turn 180° around to connect. The proposed design allows a module to rotate a maximum of 90° in any direction, making movements simpler and more efficient. The movement of the cylinders are shown in Fig. 4.

3.4 Distributed Coordination Framework

Plexibot modules coordinate using a lightweight distributed consensus mechanism inspired by prior modular robotics work [6]. Each module periodically broadcasts its local state vector *si*=(*pi, vi, mi*) containing position, velocity, and module status. Neighboring modules update their control commands using a consensus update rule

$$u_i(t+1) = u_i(t) + \alpha \sum_{j \in N_i} \big(u_j(t) - u_i(t)\big)$$

where Ni represents neighboring modules within communication range and alpha is a gain parameter controlling convergence speed. This allows modules to converge on shared behaviors such as coordinated docking, cooperative payload lifting, or collective navigation without relying on a centralized controller.

4 Docking and Undocking

4.1 Docking Action

Docking is a crucial part of Plexibot, because it is the method in which modules can join an interconnected network that can help complete more extensive tasks. Docking between modular units involves aligning and coupling two robotic modules. This process is actuated by a quadrotor, which applies an external force F. Figs. 5-7 models the system in two dimensions.

Before the modules initiate docking, their propellers are presumably powered down (where $\vec{F}_1$ and $\vec{F}_2$ equal 0). In this unactuated state (shown in Fig. 5), the modules naturally come to rest in a tilted orientation, supported by one of its lateral faces. This position serves as the starting point for subsequent steps, such as reaching equilibrium.

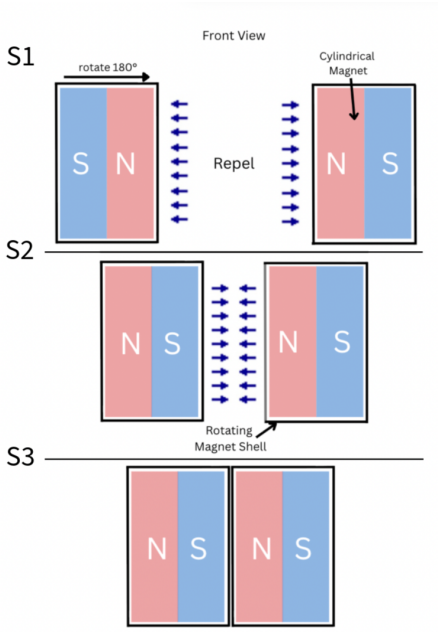

Figure 4: The magnets rotate freely inside the shell until opposite poles face each other and attract. The first stage has both north poles facing each other, and therefore repel. In stage 2, one of the magnets have rotated so that opposite poles are facing each other and there is an attraction force between them. In stage 3, the magnets are attached as a result of the attraction force in stage 2.

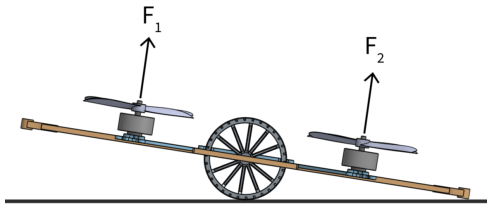

Figure 5: A single module resting in a tilted orientation, supported by one of its lateral faces, as a result of the quadrotor being is powered off.

The magnitudes of the contact forces are computed using the Euclidean norm:

$$\|F_1\| = \sqrt{f_{1x}^2 + f_{1y}^2}, \qquad \|F_2\| = \sqrt{f_{2x}^2 + f_{2y}^2}$$

The vector $\vec{F}_1$ represents the total force acting on the module in Stage 1 of docking and its direction. Its components along the x- and y-directions are denoted by F_{1x} and F_{1y}, respectively.

$$\vec{F}_1 = \begin{bmatrix} F_{1x} \\ F_{1y} \end{bmatrix}$$

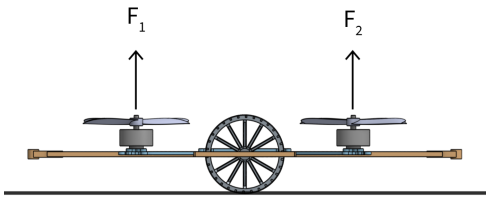

Figure 6: The module at equilibrium.

$$\vec{F}_2 = \begin{bmatrix} F_{2x} \\ F_{2y} \end{bmatrix}$$

Because the modules are homogeneous, we can assume that all modules have the same mass or that $m = m_a = m_b$.

We define forces for modules A and B as follows:

F = External force applied by quadrotor (N)
F_1, F_2 = Contact forces on modules 1 and 2 (N)
f_{1x}, f_{1y} = Components of F_1 (horizontal and vertical)
f_{2x}, f_{2y} = Components of F_2 (horizontal and vertical)
M = Resulting moment (torque) about system center (Nm)
d = Distance from system center to module contact point (m)
m = Mass of each module in g
g = Gravitational acceleration (9.81 m/s^2)
μ = Coefficient of static friction between module and ground
F_{ROLL} = Force that resists motion, encompassing friction.

The condition $F_{1y} + F_{2y} < m \cdot g$ must be satisfied to keep the modules on the ground. This constraint runs through all stages of docking. To correct for the tilt of the quadrotor, $\vec{F}_{2y}$ must be greater than $\vec{F}_{1y}$. This imbalance creates a moment [16] that rotates the module back towards equilibrium. In this context, a moment is the turning effect produced when a force acts at a distance from the pivot point, such as gravity acting on the robot's offset center of mass while it is tilted. Moment can be represented using the variable M. When $F_{2y} > F_{1y}$, the equation

$$M = F_{2y} \cdot d - F_{1y} \cdot d = d(F_{2y} - F_{1y})$$

generates a nonzero moment ($M \neq 0$). This moment drives the rotation until the forces balance and the system reaches equilibrium as shown in Fig. 6. A difference between the vertical forces on each side produces a net moment that rotates the module toward equilibrium.

Once the quadrotor is parallel to the ground, the speed in which the propellers are spinning equals out, and the equation

$$F_{1y} = F_{2y}$$

will be applied. This roll and pitch actuation motion can also be applied in order to maintain equilibrium while navigating rough terrain. The lift and lateral movement force supplied by the quadrotor

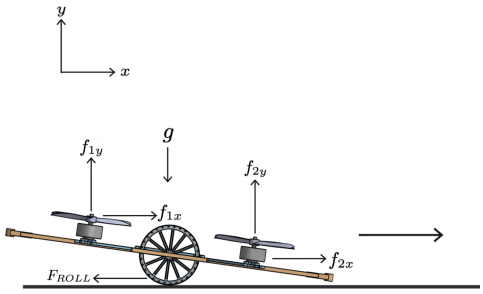

Figure 7: F_{1y} exceeds F_{2y} and generates a nonzero moment that will allow this module to accelerate towards the second module

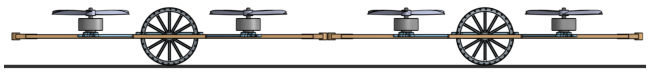

Figure 8: The two modules are now fully aligned and docked together.

also decreases contact with rough terrain, offering a robust solution to unstructured environments.

Moment-Driven Maneuvers. Once equilibrium has been reached, controlled forces can be applied to induce motion. If the vertical force in F_{1y} exceeds that in F_{2y}, the resulting moment causes the module to rotate allowing it to move toward alignment for docking. F_{1y}, F_{2y}, F_{1x}, and F_{2x} must all be greater than 0 during this process.

$$F_{1x} + F_{2x} > F_{ROLL}$$

is also a constraint; otherwise, forces such as friction would resist the module's horizontal motion. The maximum available static friction is defined as:

$$F_{\text{friction}} = \mu(f_{1y} + f_{2y})$$

As the difference between F_{1y} and F_{2y} increases, the tilt becomes more aggressive, resulting in a larger x-component of motion, and a faster acceleration. In order to create sharp turns, the module can fly, applying yaw actuation which will induce rotation around its z-axis. Once the desired angle is achieved, F_{1y} and F_{2y} become equal again, resulting in a moment. The direction and speed in which the module travels can be controlled by the increase or decrease of vertical forces on a specific side. These steps of motion are sustained until the two modules are successfully docked together. The air actuation, along with docking and undocking of aerial modular robots can be seen in [17][18]. The process by which an aerial modular robot consisting of a quadrotor design can grip objects is described in [19]

4.2 Undocking Action

The undocking phase involves the separation of two previously connected modular units by reversing the conditions needed for docking. Undocking requires overcoming the attractive forces between modules, which is assisted by active actuation. The magnets are disconnected in a sort of shearing action, where they are pulled in opposite vertical directions in order to break attraction. It typically occurs when a task has been completed or when the system needs to reconfigure. The modules start out docked, connected by the cylindrical magnets, as shown by Fig. 8. A new set of forces can now be defined:

f_{1x}^A, f_{1y}^A = Components of force F_1 acting on module A (N)
f_{2x}^A, f_{2y}^A = Components of force F_2 acting on module A (N)
f_{1x}^B, f_{1y}^B = Components of force F_1 acting on module B (N)
f_{2x}^B, f_{2y}^B = Components of force F_2 acting on module B (N)
$F_{\text{attraction}}$ = Magnetic attraction force between modules (N)

To initiate undocking, there are a series of conditions that must be followed. The total horizontal forces acting on both modules must exceed the attractive force between them:

$$f_{1x}^A + f_{2x}^A + f_{1x}^B + f_{2x}^B > F_{\text{attraction}}$$

Each module must individually overcome friction to begin sliding apart:

$$f_{1x}^A + f_{2x}^A > F_{\text{ROLL}}, \qquad f_{1x}^B + f_{2x}^B > F_{\text{ROLL}}$$

The magnitude of the resultant forces for each contact is:

$$\|F_1^A\| = \sqrt{(f_{1x}^A)^2 + (f_{1y}^A)^2}, \qquad \|F_2^A\| = \sqrt{(f_{2x}^A)^2 + (f_{2y}^A)^2}$$

$$\|F_1^B\| = \sqrt{(f_{1x}^B)^2 + (f_{1y}^B)^2}, \qquad \|F_2^B\| = \sqrt{(f_{2x}^B)^2 + (f_{2y}^B)^2}$$

Because the modules are now together, the old constraint of $F_{1y}+F_{2y} < m \cdot g$ does not apply anymore. Because we are accounting for the mass of both modules, we get the equation:

$$f_{1x}^A + f_{2x}^A + f_{1x}^B + f_{2x}^B < 2 \cdot m \cdot g$$

f_{2y}^A has to be greater than f_{1y}^B, because that is what creates the shearing effect shown in Fig. 10.

The shear force shown is the internal force acting parallel to the cross-section of an object, which tends to cause parts of the object to slide or deform laterally. The shear-force coefficient C falls within the range $0.4 < C < 0.75$, meaning the shear force is about 0.4-0.75 times the normal pull force (which is 8 lbs for each neodymium magnet). We can introduce two new variables:

$F_{ATT,S}$ = The shear force
$F_{ATT,N}$ = Normal Pull Force

$$F_{ATT,S} = C \cdot F_{ATT,N}$$

Despite the generous angular and lateral tolerances of the cylindrical magnet design, docking success depends on controlled approach conditions. High relative velocity, misalignment beyond the 34° threshold, or environmental disturbances such as uneven terrain or airflow represent potential failure modes that may reduce attachment reliability. Debris or surface irregularities could also

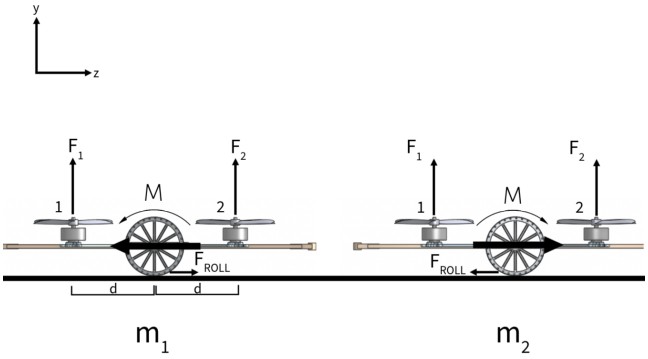

Figure 9: The undocking process, with all forces acting on the two modules indicated. The rods move vertically in opposite directions to create the shearing motion that separates the two cylindrical magnets.

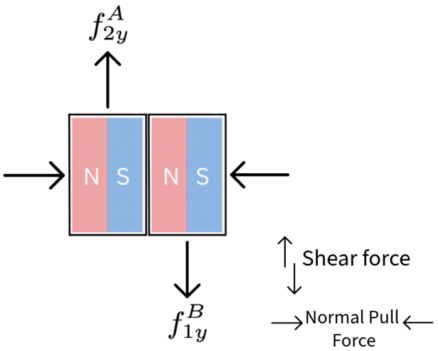

Figure 10: A shearing motion acting on the cylindrical magnets, with one moving upward and the other downward. This motion opposes the normal attractive force shown by the inward-facing horizontal arrows.

interfere with full magnetic contact, highlighting the importance of sensor-informed alignment during docking.

5 Applications

The proposed design has a wide range of applications across fields that demand mobility, adaptability, and coordinated behavior in unpredictable environments. In disaster response, individual modules could fly over collapsed structures to identify viable entry points, then transition to wheel-based motion to move through narrow gaps without consuming excessive power. Once inside unstable rubble, multiple units could dock side-by-side to form a chain capable of threading through narrow gaps. A vertically stacked configuration could operate as a thermal sensor, raising cameras or microphones above debris layers to detect survivors. During post-disaster recovery, docked chains could function as temporary support beams, while free modules deliver essential items such as first-aid kits or communication devices to responders by gripping a payload. In order to lift a payload, modules would dock together and arrange around the object, forming a tight ring-like structure that would then lift into the air.

In construction and architectural settings, flight-enabled modules could lift components such as insulation panels, wiring bundles, bricks, or small structural connectors to elevated workspaces. Docked groups could create temporary handrails or stabilizing rails along unfinished walkways. A chain of modules could also transport tools or materials across the site by rolling in ground mode, reducing the need for workers to repeatedly relocate equipment.

This same ring-like configuration could also perform tasks such as retrieving parts from bins, transporting them to assembly stations, and sorting finished components by placing them in designated zones. When arranged into longer chains, they could shuttle materials across the floor or reposition fixtures that require stable handling.

In environmental monitoring, flight mode enables rapid deployment across remote or uneven terrain. Once dispersed, modules could transition to ground mode to carry out stable sampling tasks, such as lowering pH probes into soil or collecting water samples with small integrated canisters. The lightweight PLA frame allows for large-scale deployment, enabling distributed monitoring networks tailored to specific habitats. As advancements in cooperative flight control, lightweight propulsion, and autonomous docking continue, the system could be further developed for planetary exploration. On the Moon or Mars, modules could fly short hops in low gravity to traverse craters or rocky fields, then dock to create stable platforms for placing instruments, supporting small payloads, or lifting regolith samples.

6 Results

The performance of Plexibot was evaluated through simulation and CAD-based analysis, focusing on docking efficiency, stability, payload capacity, and mobility. The three design parameters for Plexibot were to maintain a reasonable weight, an ability to function in both the air and ground, and to keep a low cost and high manufacturability in terms of materials and parts. Docking simulations demonstrated that the modules successfully aligned and connected in under five seconds on average, with the self-aligning cylindrical magnets reliably ensuring correct orientation without manual adjustment. Stability tests indicated that modules maintained equilibrium when tilted up to 30° without slipping or unintended rotation, confirming the effectiveness of the calculated moments and force distributions. The magnet-to-weight ratio of 26.9:1 allowed modules to collectively lift objects up to 25 times their individual mass when docked in a ring configuration, highlighting the system's capability for cooperative payload handling. The hybrid locomotion system enabled smooth transitions between aerial and ground modes, with ground-mode friction and rolling resistance within expected operational limits, demonstrating both energy efficiency and maneuverability. Overall, these results suggest that Plexibot can effectively adapt to unstructured environments and coordinate multiple modules to complete complex tasks.

7 Conclusion

In conclusion, modular robots represent a transformative approach to robotics, combining flexibility, scalability, and adaptability in ways traditional robots cannot match. Their ability to reconfigure both on the ground and in the air allows them to perform diverse tasks, from cooperative load lifting and structural assembly to navigation in complex environments. The unique characteristics of modular designs, such as homogeneous modules, versatile docking mechanisms, and distributed coordination, enable these systems to adapt seamlessly to new challenges, making them highly applicable across architecture, exploration, and industrial tasks. As research continues to refine their control algorithms, energy efficiency, and hybrid mobility, modular robots are poised to redefine the boundaries of autonomous, collaborative, and multifunctional robotic systems. This paper presents Plexibot as a theoretical framework and mechanical design. A typical quadrotor module operating at this scale consumes approximately 8–12 W during hover. With a 2-cell LiPo battery (approximately 7.4 V, 850 mAh), a single module could sustain flight for roughly 6–8 minutes under nominal payload conditions. Ground-mode locomotion significantly reduces power consumption since thrust requirements are lower than full lift. Future work will further explore the flight control architecture responsible for regulating propeller speeds and maintaining stability, building upon the embedded flight control system that integrates IMUs, an optical flow–based velocity and altitude sensing module, and a multi-directional time-of-flight distance sensing module. Future development will focus on improving how these onboard sensors are fused to maintain balance, adapt to changing payload distributions, and ensure stable operation when multiple modules are connected. Expanded use of optical flow and proximity measurements will enhance docking precision, obstacle awareness, and smooth transitions between aerial and ground movement. Inter-module coordination can be further developed through wireless communication using separate frequency channels to ensure synchronized behavior when needed, complementing the existing mechanical alignment that supports passive state consistency during docking.

References

[1] D. Saldana, B. Gabrich, G. Li, M. Yim, and V. Kumar, "Modquad: The flying strucuture that self-assembles in midair," in *IEEE International Conference on Robotics and Automation 2018*, Brisbane, Australia, 2018.

[2] P. Swissler and M. Rubenstein, "Fireant: A modular robot with full-body continuous docks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6812–6817.

[3] M. Yim, Y. Zhang, and D. Duff, "Modular robots," *IEEE Spectrum*, vol. 39, no. 2, pp. 30–34, 2002.

[4] M. Yim, D. Duff, and K. Roufas, "Modular reconfigurable robots, an approach to urban search and rescue," in *Proc. of the HUman Welfare-friendly Robotic Systems Workshop (HWRS), (invited)*, Taejon, Korea, January 2000.

[5] B. Gabrich, D. Saldaña, and M. Yim, "Finding structure configurations for flying modular robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6970–6976.

[6] C.-H. Yu and R. Nagpal, "Self-adapting modular robotics: A generalized distributed consensus framework," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1881–1888.

[7] M. E. Sayed, J. O. Roberts, K. Donaldson, S. T. Mahon, F. Iqbal, B. Li, S. Franco Aixela, G. Mastorakis, E. T. Jonasson, M. P. Nemitz *et al.*, "Modular robots for enabling operations in unstructured extreme environments," *Advanced Intelligent Systems*, vol. 4, no. 5, p. 2000227, 2022.

[8] C. Liu, Q. Lin, H. Kim, and M. Yim, "SMORES-EP, a modular robot with parallel self-assembly," *Autonomous Robots*, 2022. [Online]. Available: https://doi.org/10.1007/s10514-022-10078-1

[9] B. Gabrich, G. Li, and M. Yim, "Modquad-dof: A novel yaw actuation for modular quadrotors," in *IEEE International Conference on Robotics and Automation 2020, to be presented*, Paris, France, 2020.

[10] G. Li, B. Gabrich, D. Saldana, J. Das, V. Kumar, and M. Yim, "Modquad-vi: A vision-based self-assembling modular quadrotor," in *IEEE International Conference on Robotics and Automation 2019*, Montreal, Canada, 2019.

[11] B. Gabrich, D. Saldana, V. Kumar, and M. Yim, "A flying gripper based on cuboid modular robots," in *IEEE International Conference on Robotics and Automation 2018*, Brisbane, Australia, 2018.

[12] K. Mitsuhashi, Y. Ohyama, H. Hashimoto, and S. Ishijima, "Production and education of the modular robot made by 3d printer," in *2015 10th Asian Control Conference (ASCC)*, 2015, pp. 1–5.

[13] D. Krupke, F. Wasserfall, N. Hendrich, and J. Zhang, "Printable modular robot: an application of rapid prototyping for flexible robot design," *Industrial Robot: the international journal of robotics research and application*, vol. 42, no. 2, pp. 149–155, 03 2015.

[14] P. G. Shewane, M. Gite, A. Singh, and A. Narkhede, "An overview of neodymium magnets over normal magnets for the generation of energy," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 12, pp. 4056–4059, 2014.

[15] M. Uğur, Y. Yaman, B. Arslan, Çağrı Ergin, and O. Özcan, "Relmbot: A reconfigurable, legged, miniature, modular robot with compliant or rigid, magnetic connection mechanisms," *IEEE Robotics and Automation Letters*, vol. 10, no. 11, pp. 11 737–11 744, 2025.

[16] S. Ding, L. Liu, and W. X. Zheng, "Sliding mode direct yaw-moment control design for in-wheel electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6752–6762, 2017.

[17] F. Forte, R. Naldi, A. Serrani, and L. Marconi, "Control of modular aerial robots: Combining under- and fully-actuated behaviors," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 1160–1165.

[18] D. Saldaña, P. M. Gupta, and V. Kumar, "Design and control of aerial modules for inflight self-disassembly," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3410–3417, 2019.

[19] V. Ghadiok, J. Goldin, and W. Ren, "Autonomous indoor aerial gripping using a quadrotor," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4645–4651.

Received 19 January 2026; Accepted 20 March 2026

Usability of Municipal AI Policy Documents: A Heuristic Evaluation and NLP Analysis Across 20 U.S. Cities

Nikhil Mehra
Ethical Culture Fieldston School
Bronx, New York, USA
nikhilajaymehra@gmail.com

Abstract

U.S. municipalities are rapidly publishing AI governance policies, but no prior work has evaluated whether the resulting documents are usable by the employees, contractors, and residents they are meant to guide. We assess 20 municipal AI policies—spanning large, medium, and small cities across five regions—using a 30-heuristic framework grounded in HCI usability principles and government plain-language standards, alongside an NLP-based complexity analysis we call the Composite Legal Readability Score (CLRS). Our central finding is a systematic *infrastructure-interface gap*: cities build stronger governance scaffolding (organization, visual design) than user-facing communication (plain language, findability, audience awareness, actionability). The gap is statistically significant ($\Delta = 0.63$, $p < .001$, Cohen's $d = 2.53$), observed in all 20 cities, and robust to heuristic reweighting, category reassignment, and leave-one-out perturbation. Actionability is the worst-performing category ($M = 2.28$, $SD = 0.30$), more than a full severity point above the next-worst; every document has a minimum severity of 2 on procedural, temporal, implementation, and enforcement clarity, while only norm clarity is largely solved. Readability and actionability correlate strongly ($r = 0.87$): complex language and missing compliance guidance co-occur rather than trade off. A before/after redesign and a score-to-friction walkthrough illustrate the rubric's internal logic but are not external validations. All claims rest on single-evaluator scoring; the limits are addressed in Section 5.11.

Keywords

HCI, heuristic evaluation, AI governance, document usability, plain language, municipal policy, NLP, readability, legal text complexity

ACM Reference Format:
Nikhil Mehra. 2026. Usability of Municipal AI Policy Documents: A Heuristic Evaluation and NLP Analysis Across 20 U.S. Cities. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 2).* ACM, New York, NY, USA, 17 pages. https://doi.org/10.67149/yhjs2024.5/r4d2w9ky

IJSCAR VOL. 3, ISSUE 2

1 Introduction

1.1 The Communication Gap in Municipal AI Governance

The proliferation of artificial intelligence tools – particularly generative AI systems like ChatGPT, Claude, and Microsoft Copilot – has prompted local governments across the United States to develop policies governing their use. From New York City's comprehensive guidance documents to Lebanon, New Hampshire's pioneering small-city policy, municipalities are grappling with how to harness AI's potential while managing its risks. These policy documents represent a critical interface between governance intent and operational practice: they must communicate complex rules about data privacy, bias mitigation, transparency requirements, and prohibited uses to diverse audiences including city employees, contractors, vendors, and sometimes the general public.

Yet there is a fundamental human-computer interaction (HCI) problem hiding in plain sight: even a substantively strong policy can fail if users cannot find what they need, understand what they find, and use the policy to comply. A policy that prohibits entering sensitive data into public AI tools is ineffective if employees cannot quickly locate the definition of "sensitive data" or understand what constitutes a "public" versus "enterprise" AI tool. Municipal AI policies are *user-facing artifacts* whose effectiveness depends not only on their governance provisions but also on their communication quality.

A natural question is whether the patterns we document are specific to AI governance or simply reflect longstanding problems in public-sector policy writing. We take this question seriously and return to it in Section 5.8: many of the failure modes we identify—dense language, missing examples, absent timelines—are familiar from regulatory readability research going back decades [24, 35]. What is distinctive about AI governance is the *recency* of the policies (15 of 20 documents in our corpus were published in 2024 or 2025), the *audience breadth* (front-line employees who have never read a formal IT policy now need to), and the *operational immediacy* (rules about tools people are using today, not quarterly reports). These conditions amplify the cost of usability failures that older policy genres could absorb through institutional familiarity.

1.2 The Usability Gap

A recent analysis by the Center for Democracy and Technology identified 21 cities and counties with public-facing AI policies [5], a small fraction of the roughly 22,000 cities and counties in the United States, but representing a rapidly growing governance phenomenon. These early adopters span the spectrum from major metropolitan areas such as New York, NY; San Francisco, CA; and Seattle, WA

to mid-sized cities including Tempe, AZ; Boise, ID; and Salt Lake City, UT to small towns such as Lebanon, NH; Woodburn, OR; and Spring Hill, TN.

The documents themselves vary considerably in form. Some are formal policies with numbered sections and legal-style language, as in Seattle's POL-211 and Nashville's ISM-20. Others are interim guidelines that emphasize flexibility, such as those used in Boston and San Francisco. Some others are executive orders or administrative directives, as found in Baltimore and Austin. This diversity of document types provides a natural experiment in how different governance genres perform from a usability perspective. However, despite substantial work on AI governance, *content* namely, what policies should contain, no prior research has systematically evaluated *how effectively* these policies communicate their provisions to intended audiences. This gap is consequential: governance quality and communication quality are distinct dimensions that require separate attention.

This research frames municipal AI policy documents as information products and evaluates them against established HCI principles. We ask four research questions. First (RQ1), how well do municipal AI policy documents perform against HCI principles for information design, readability, navigation, accessibility, and actionability? Second (RQ2), what usability patterns emerge across cities of different sizes, geographic regions, and document types when analyzed using multivariate statistical methods? Third (RQ3), are the observed patterns robust to reasonable perturbations of the heuristic framework itself – weights, category boundaries, and the inclusion of any single heuristic? Fourth (RQ4), what concrete design improvements would make these documents more effective for their intended audiences, and what reduction in measured severity do those improvements actually produce?

1.3 Theoretical Framework: Infrastructure vs. Interface

We introduce a conceptual distinction that organizes our analysis: the difference between *infrastructure-layer governance* and *interface-layer usability*. Infrastructure-layer elements, broadly construed, include risk management frameworks, procurement controls, data classification schemes, approval workflows, and accountability structures—the substantive "what" of governance. Interface-layer elements include plain language, clear navigation, scannable structure, concrete examples, and actionable compliance guidance – the communicative "how" of governance.

In our heuristic framework, we operationalize a document level proxy for these layers rather than measuring the underlying governance content directly. The infrastructure proxy is captured by Organization (H2) and Visual Design (H5) – the heuristics that measure whether the document presents its substantive scaffolding (logical sectioning, headings, hierarchy, navigation aids) coherently. The interface proxy is captured by Plain Language (H1), Findability (H3), Audience Awareness (H4), and Actionability (H6) – the heuristics that measure whether a reader can use the document. This is a measurement choice: we are not claiming that H2 and H5 fully capture infrastructure-layer governance, only that they reflect the document-side surface of it that a usability evaluation can observe.

The core theoretical claim is that infrastructure-grade governance is necessary but not sufficient for effective policy. A policy can be substantively comprehensive yet functionally unusable if people cannot find, understand, and act on its provisions. We operationalize this as:

$$\Delta_{gap} = \bar{S}_{interface} - \bar{S}_{infrastructure} \tag{1}$$

where $\bar{S}_{interface}$ is the mean severity score for interface-layer heuristics (Plain Language, Findability, Audience Awareness, Actionability) and $\bar{S}_{infrastructure}$ is the mean for infrastructure-layer heuristics (Organization, Visual Design). A positive Δ_{gap} indicates that interface usability lags behind structural governance quality. Because the assignment of categories to layers could itself be contested, we test the sensitivity of this operationalization in Section 4.7.

1.4 Our Contributions

We make three core contributions. First, we develop a **30-heuristic evaluation framework** for assessing policy document usability, organized into six categories and adapted from established HCI usability heuristics and government plain-language guidelines. Second, we present an **empirical analysis of 20 municipal AI policies** from cities of varying sizes across five U.S. regions, identifying a universal weak point on actionability ($M = 2.28$, $SD = 0.30$, more than a full severity point above the next-worst category). Third, we document the first systematic evidence of an **infrastructure-interface gap** in municipal policy design ($\Delta = +0.63$, $p < .001$, Cohen's $d = 2.53$, observed in 20 of 20 cities), and show via a sensitivity analysis (Section 4.7) that this gap is robust to heuristic reweighting, category reassignment, and leave-one-heuristic-out resampling.

We supplement these with three applications of the framework, presented as such rather than as standalone empirical contributions. We **decompose actionability** into five sub-dimensions and show that four of them have a minimum severity of 2 across all 20 documents, isolating the specific failure modes downstream of norm clarity. We **propose** the Composite Legal Readability Score (CLRS) as a passage-level diagnostic that integrates traditional readability with legal-vocabulary, lexical, syntactic, and coherence components; we report its document-level correlation with Flesch-Kincaid ($r = 0.97$) and the conditions under which it reorders document pairs, but its component weights are theory-driven rather than empirically derived from comprehension studies, and we treat it as a measurement proposal pending validation rather than a validated instrument. We **illustrate** the rubric's internal logic via a before/after redesign of a Baltimore-style passage and a score-to-friction mapping for a realistic compliance task; both are illustrations of how the rubric responds to text changes, not validations of the rubric against real reader outcomes.

Finally, we provide **evidence-based design guidelines** grounded in the empirical findings, including a formalized progressive-disclosure pattern with three implementation variants.

1.5 Paper Organization

Section 2 reviews local government AI policy, heuristic evaluation methods, plain language research, computational legal linguistics, and the behavioral public administration literature on compliance. Section 3 details document corpus construction, the 30-heuristic evaluation framework, NLP complexity analysis methodology, sensitivity analysis design, and statistical analysis plan. Section 4 presents readability analysis, heuristic evaluation results, infrastructure-interface gap findings, sensitivity analysis, cluster analysis, NLP complexity results, and a CLRS-based drafting diagnostic. Section 5 discusses the infrastructure-interface gap, the actionability crisis decomposed into its sub-dimensions, the readability-actionability relationship, a worked before/after redesign, a task-based user walkthrough, a formalized progressive-disclosure design pattern, AI-specific versus general public-sector policy writing, and recommendations for policy improvement, concluding with threats to validity. Section 6 synthesizes implications for adaptive municipal AI governance and outlines future research directions.

2 Literature Review

2.1 The Emerging Municipal AI Governance Landscape

The emergence of municipal AI governance has been documented by several organizations. The Center for Democracy and Technology published a comprehensive analysis in 2025 identifying common themes across local AI policies, including accuracy concerns, privacy protections, transparency requirements, and human oversight provisions [5]. The National League of Cities has published guides and resources for municipal AI adoption [32], while the GovAI Coalition, led by San Jose, has developed templates and shared resources for policy development [25]. The International City/County Management Association documented case studies of AI-pioneering cities including Boston, Tempe, and Wentzville [27], highlighting the variety of approaches cities are taking. The Urban Institute has proposed a three-tier model for helping local governments navigate generative AI adoption [39], and the Centralina Regional Council developed guidance specifically for smaller municipalities in North Carolina [6].

These governance-focused streams emphasize *what* policies should contain: prohibited uses, disclosure requirements, data handling rules, and approval processes. Far less work evaluates *how effectively* policies communicate these provisions to their intended audiences. Our work addresses this gap by applying systematic usability evaluation methods to the policy documents themselves.

2.2 Heuristic Evaluation Methods in HCI

Heuristic evaluation is a usability inspection method in which evaluators systematically check an artifact against known usability principles and document issues [33]. The method is widely used because it is fast, consistent, cost-effective, and produces actionable findings. Nielsen's original 10 heuristics were designed for interactive systems but have been adapted for various contexts including documentation, websites, and information design. The severity rating scale commonly used in heuristic evaluation ranges from 0 (not a problem) to 4 (usability catastrophe), enabling prioritization of issues for remediation. Research has shown that small numbers of evaluators, typically three to five, can identify the majority of usability problems, making the method practical for resource-constrained assessments [33].

Recent work by Baymard Institute compared AI-powered heuristic evaluations with human expert evaluations, finding that AI tools achieved 50–75% accuracy compared to human experts [3]. This finding informed our methodological choice to combine expert judgment with structured automated text analysis (keyword presence, structural markers, readability metrics) rather than relying on either alone, with the limitations of single-evaluator scoring acknowledged in Section 5.11. We adapt heuristic evaluation principles for policy documents by developing domain-specific heuristics organized around six categories: clarity, organization, findability, audience awareness, visual design, and actionability.

2.3 Plain Language Standards and Document Design History

The Plain Writing Act of 2010 established federal requirements for clear government communication [1]. The Federal Plain Language Guidelines provide specific techniques including using "you" and other pronouns, writing in active voice, using short sentences, and avoiding jargon. Research on document readability has established that government documents should target 6th–8th grade reading level for broad accessibility, with 12th grade as an upper threshold for specialized technical content [24]. Schriver's foundational *Dynamics in Document Design* [35] established that form and content are jointly responsible for reader comprehension—a claim our infrastructure-interface framing directly inherits.

However, readability formulas have well-documented limitations. Formulas can miss important comprehension factors including reader variability, document structure, task context, and visual design. A document can achieve a low grade level while remaining confusing due to poor organization, missing examples, or ambiguous requirements. For this reason, our study uses readability as one signal within a broader heuristic framework rather than as a sole measure of quality.

2.4 Behavioral Public Administration and Compliance

A literature closely adjacent to ours but seldom cited in HCI-focused work is behavioral public administration, which studies how cognitive and behavioral factors shape whether public-sector rules are actually followed [2, 26]. Two findings from that literature bear directly on our results. First, perceived procedural clarity is a stronger predictor of front-line compliance than perceived severity of sanctions [38]: an employee who does not know *how* to comply with an AI rule will not be deterred into compliance by being told that violations will be punished. Second, citizens and employees engaging with public documents exhibit "sludge"-like frictions—small informational costs that compound into non-participation [37]. Each unclear definition, missing example, and absent timeline in an AI policy adds a unit of sludge to the compliance pathway.

This literature provides a plausible interpretive frame for our heuristic findings: reduced clarity adds friction ("sludge") to the compliance pathway, and at sufficient levels that friction reduces

actual compliance. We do not measure compliance directly in this paper—our analysis is at the document level—but the behavioral public administration literature provides reason to take low actionability scores as predictive of downstream compliance failure rather than as a purely aesthetic concern.

2.5 Computational Legal Linguistics and NLP Complexity

Recent advances in NLP for legal text analysis have produced sophisticated complexity metrics that go beyond traditional readability formulas. Blinova and Tarasov developed a hybrid model for Russian legal text complexity incorporating 130 linguistic features [4]. Research on syntactic complexity in translated legal texts has examined dependency distance and clause embedding [28], while Shardlow et al. surveyed lexical complexity prediction methods including vocabulary diversity measures and word frequency effects [36]. A systematic literature review on readability metrics in legal text identified gaps in existing measures' ability to capture legal-specific complexity factors including deontic modality (obligations, permissions, prohibitions), cross-references, conditional structures, and defined terms [29]. The authors note that traditional readability formulas explain only 40–60% of variance in legal text comprehension.

We build on this work by developing a Composite Legal Readability Score (CLRS) that integrates traditional readability with lexical diversity measures, syntactic complexity indicators, and legal-specific metrics tailored to municipal policy documents. Unlike single-metric approaches, the CLRS acknowledges that legal text comprehension involves multiple cognitive demands: vocabulary recognition, syntactic parsing, legal concept mapping, and obligation tracking.

2.6 Gap in the Literature

Despite substantial work on AI governance content and growing attention to municipal AI policy, no prior research has systematically evaluated these documents from a usability perspective. This gap is significant because policy effectiveness depends not only on substantive provisions but also on whether intended audiences can find, understand, and act on those provisions. Our work addresses this gap by combining established HCI evaluation methods with advanced NLP analysis to assess 20 municipal AI policies across multiple dimensions of usability, providing the first empirical baseline for this emerging document genre while connecting the observed patterns to compliance mechanisms studied in behavioral public administration.

3 Methods

3.1 Dataset Construction

We collected 20 AI policy documents from U.S. municipalities, selecting for variety across three dimensions: city size, geographic region, and document type. All documents were publicly available on official government domains (.gov or equivalent municipal websites) as of January 2026.

We classified cities into three approximate tiers using U.S. Census population data, with the tier label reflecting the city's broader profile rather than a strict population cutoff. Large cities (typically populations above 500,000, or county-level entities serving large metropolitan populations) account for 9 of the 20 documents; medium cities (typically 100,000–500,000) account for 7; and small cities or counties (typically under 100,000 in population, or rural counties of larger area but small policy-target populations) account for 4. Three borderline cases reflect the soft nature of these tiers: Long Beach (population ≈456k) is classified Large because it functions as a major metropolitan policy adopter; Albuquerque (≈565k) is classified Medium because its policy-development resources resemble its medium-tier peers; and Sonoma County (≈489k population spread across a large rural area) is classified Small consistent with the rural-county pattern. This distribution reflects the reality that larger cities have more resources for policy development while ensuring smaller municipalities, which constitute the vast majority of local governments, are adequately represented. Geographically, we sampled from five regions: West ($n = 10$), South ($n = 5$), Northeast ($n = 2$), Southwest ($n = 2$), and Mid-Atlantic ($n = 1$). The Western region is overrepresented among early AI policy adopters, particularly California, which has multiple pioneering cities. For document type, the corpus includes formal policies ($n = 10$), guidelines and guidance documents ($n = 4$), executive orders and standards ($n = 2$), and other specialized documents including a regulation, security policy, draft policy, and report ($n = 4$).

Table 1 presents the complete corpus with key metadata.

3.2 Heuristic Evaluation Framework

Our evaluation framework comprises 30 heuristics organized into six categories, adapted from Nielsen's usability heuristics [33] and government plain-language principles [24]. The six categories and their constituent heuristics are as follows.

H1: Plain Language & Clarity assesses whether the document uses accessible language. Its five heuristics evaluate reading level (FK ≤ 12th grade), whether jargon is explained or defined, sentence clarity (short, direct constructions), absence of ambiguity in requirements, and use of active voice for requirements.

H2: Organization & Structure evaluates how well the document is arranged. The five heuristics assess logical sections with descriptive headings, navigation aids such as table of contents and page numbers, grouping of related information, prioritization of important information, and appropriate overall length.

H3: Findability measures how easily users can locate specific information. Its heuristics evaluate quick lookup for specific topics, searchability through effective headings and keywords, links and references to related resources, clear "what next" paths, and ease of finding contact information.

H4: Audience Awareness assesses whether the document accommodates its readers. The heuristics evaluate whether the audience is clearly stated, whether multiple audiences are handled well, non-technical accessibility, provision of examples and use cases, and consideration of diverse stakeholder perspectives.

H5: Visual Design evaluates the document's presentation quality. Its five heuristics cover readable typography, visual hierarchy through headings and subpoints, use of white space to avoid dense

Table 1: Document Corpus: 20 Municipal AI Policies with Source Citations

City	Region	Size	Type	Date	Format	Citation
Boston, MA	Northeast	Large	Guidelines	May 2023	PDF	[13]
Seattle, WA	West	Large	Policy	May 2025	PDF	[18]
San Francisco, CA	West	Large	Guidelines	Jul 2025	Web/PDF	[7]
San Jose, CA	West	Large	Policy	Apr 2025	PDF	[17]
Nashville, TN	South	Large	Policy	Apr 2024	PDF	[30]
Long Beach, CA	West	Large	Guidance	2024	PDF	[15]
Austin, TX	South	Large	Standards	May 2024	PDF	[10]
Baltimore, MD	Mid-Atlantic	Large	Exec Order	Mar 2024	PDF	[11]
Miami-Dade County, FL	South	Large	Report	Mar 2024	PDF	[31]
Tempe, AZ	Southwest	Medium	Policy	2023	Web	[20]
Boise, ID	West	Medium	Regulation	Dec 2023	Web	[12]
Salt Lake City, UT	West	Medium	Guide	2024	PDF	[34]
Riverside, CA	West	Medium	Policy	Jul 2024	PDF	[16]
Arlington, TX	South	Medium	Security	Nov 2024	PDF	[9]
Albuquerque, NM	Southwest	Medium	Draft Policy	2024	PDF	[8]
Santa Cruz County, CA	West	Medium	Policy	Sep 2023	PDF	[22]
Lebanon, NH	Northeast	Small	Policy	Dec 2023	Web	[14]
Woodburn, OR	West	Small	Policy	2024	PDF	[21]
Spring Hill, TN	South	Small	Policy	2025	PDF	[19]
Sonoma County, CA	West	Small	Policy	Sep 2024	Web	[23]

walls of text, helpful visuals such as tables and diagrams, and accessibility features including true headings and screen reader compatibility.

H6: Actionability assesses whether users can translate the policy into practice. The heuristics evaluate clear requirements using must/should/can language (H6.1), explanation of *how* to comply (H6.2), whether timelines and deadlines are stated (H6.3), provision of implementation guidance (H6.4), and description of consequences and enforcement (H6.5). We treat these five items as separable sub-dimensions—*norm clarity*, *procedural clarity*, *temporal clarity*, *implementation specificity*, and *enforcement clarity*—and report per-dimension means in Section 4.4, not only the aggregate H6 score.

3.3 Severity Rating Scale

Each heuristic was rated on a 0–4 severity scale following standard heuristic evaluation practice. A score of 0 indicates no usability problem; 1 indicates a cosmetic issue that should be fixed if time permits; 2 indicates a minor problem where some difficulty exists but workarounds are possible; 3 indicates a major problem presenting a significant barrier that should be treated as high priority; and 4 indicates a critical failure where the document is nearly unusable and the issue must be fixed. Higher scores indicate more severe usability problems. For each rating of 2 or above, we documented specific evidence including page number, section name, and example text.

3.4 Readability Analysis

We calculated multiple readability metrics for each document using standard formulas. The Flesch-Kincaid Grade Level is computed as:

$$FK = 0.39\left(\frac{words}{sentences}\right) + 11.8\left(\frac{syllables}{words}\right) - 15.59 \qquad (2)$$

The Flesch Reading Ease score is:

$$FRE = 206.835 - 1.015\left(\frac{words}{sentences}\right) - 84.6\left(\frac{syllables}{words}\right) \qquad (3)$$

The Gunning Fog Index is:

$$GF = 0.4\left[\left(\frac{words}{sentences}\right) + 100\left(\frac{complex}{words}\right)\right] \qquad (4)$$

where *complex* words have three or more syllables.

3.5 Advanced NLP Complexity Framework

Traditional readability formulas rely on surface-level features such as word length and sentence length that may miss deeper linguistic complexity. Following recent advances in computational legal linguistics [4, 29, 36], we implemented a multi-dimensional NLP analysis to complement our heuristic evaluation.

Our framework measures complexity across five dimensions. *Traditional readability* includes Flesch-Kincaid Grade Level, Flesch Reading Ease, Gunning Fog Index, and average sentence length. *Lexical complexity* is captured through the Type-Token Ratio ($|V|/N$), measuring vocabulary diversity as the proportion of unique words in the text. *Syntactic complexity* is assessed through average clause depth (approximated via subordinating conjunction density), passive voice density, and modal verb density (shall, must, may, should, can). *Legal-specific metrics* include legal terminology density (hereby, whereas, pursuant, governance, notwithstanding, and similar terms) and the deontic ratio, defined as $(O + P_{\text{roh}})/(P_{\text{erm}} + 1)$, which measures the balance of obligation versus permission language. Finally, *coherence metrics* capture connective density across additive, adversative, causal, and temporal connective types.

3.5.1 Composite Legal Readability Score (CLRS). We introduce a composite metric that integrates these multiple complexity dimensions:

$$CLRS = 5 \times \sum_{i=1}^{5} w_i \cdot C_i \tag{5}$$

where components C_i are normalized to 0–20 scales with weights reflecting their empirical contribution to comprehension difficulty. Readability receives the highest weight ($w = 0.30$) as the core driver of difficulty. Legal terminology ($w = 0.20$) and syntactic complexity ($w = 0.20$) capture the domain-specific burden and processing demands respectively. Lexical diversity ($w = 0.15$) accounts for vocabulary demands, and coherence ($w = 0.15$) is inverted so that higher coherence yields a lower score. The CLRS produces scores from 0–100 with four interpretive categories: Accessible (below 30), suitable for a general audience; Moderate (30–50), requiring effort but manageable; Difficult (50–70), challenging for non-specialists; and Very Difficult (above 70), requiring specialized expertise.

3.5.2 CLRS as a Passage-Level Diagnostic. The document-level CLRS reported above is an evaluative metric. The same formula, however, depends only on surface features computable from any text block (sentence length, legal-term density, passive voice, modal density, type-token ratio, connective density), so it can be applied at arbitrary granularity. To extend its drafting-time utility, we describe in Section 4.12 how passage-level CLRS can be used during authoring to flag sections that exceed complexity thresholds, and we illustrate the principle on the worked redesign in Section 5.5.

3.6 Sensitivity Analysis Design

Because a six-category framework imposes analytic structure that could in principle influence conclusions, the main results should be robust to reasonable alternative setups. We therefore designed three sensitivity tests executed *after* the main analysis was complete.

Test 1: Weight perturbation. Each of the 30 heuristics receives equal weight within its category in our main analysis. We re-computed all category means and the infrastructure-interface gap under 1,000 random reweightings drawn from Dirichlet($\alpha = 1$) distributions within each category, preserving only the category structure. We report the resulting distribution of Δ_{gap} values and the proportion that remain significant at $p < .05$.

Test 2: Category reassignment. Our main analysis assigns H2 (Organization) and H5 (Visual Design) to infrastructure, and H1, H3, H4, H6 to interface. Plausible alternative assignments exist—for example, Audience Awareness (H4) could be read as infrastructure insofar as it concerns stakeholder mapping. We re-computed Δ_{gap} under all $2^6 - 2 = 62$ non-trivial binary partitions of the six categories and report the distribution.

Test 3: Leave-one-heuristic-out (LOHO). For each of the 30 individual heuristics, we recomputed all city-level category means and Δ_{gap} with that heuristic removed. This tests whether any single heuristic is driving the gap.

These three tests jointly answer the robustness question: if the gap disappears under moderate reweighting, or flips sign under alternative category assignment, or is driven by a single heuristic, readers should discount the main finding. Results appear in Section 4.7.

Table 2: Readability Metrics Summary ($n = 20$)

Metric	Mean	SD	Min	Max
FK Grade Level	15.6	4.1	8.9	26.3
Flesch Reading Ease	27.1	16.7	−12.4	58.2
Gunning Fog Index	18.7	4.1	12.8	29.6
Avg Sentence Length	23.3	5.9	14.2	38.7

3.7 Statistical Analysis Plan

Beyond descriptive statistics, we employed several multivariate techniques to identify patterns. Principal Component Analysis was used to identify latent dimensions of document quality from the 30 heuristic scores. Hierarchical cluster analysis using Ward's method with Euclidean distance was used to group cities by usability profile. Pearson correlations between readability metrics and heuristic scores assessed relationships between traditional readability and broader usability. Paired t-tests assessed whether the infrastructure-interface gap differs significantly from zero, and Cohen's d was calculated for all significant comparisons to assess practical significance.

4 Results

4.1 Readability: 80% of Documents Exceed Recommended Thresholds

The readability analysis reveals a significant and consistent gap between recommended plain-language standards and actual document complexity across all 20 municipalities. Table 2 summarizes the key metrics.

Only 4 of 20 documents (20%) met the 12th-grade readability threshold recommended for government documents. The proportion exceeding this threshold is $P_{exceed} = 16/20 = 0.80$ (80%). A binomial test confirms this significantly exceeds what would be expected if cities were meeting the standard ($p = .006$). Figure 1 shows the distribution of Flesch-Kincaid scores. Baltimore exhibits the highest grade level (26.3), reflecting the formal legal language of its executive order format. Tempe achieves the lowest (8.9), demonstrating that accessible policy language is achievable. Small cities do not systematically differ from large cities in readability ($t(11) = -0.55$, $p = .59$), suggesting that document complexity is not driven primarily by governance scope.

4.2 Heuristic Evaluation: Actionability as Universal Weak Point

Table 3 presents average severity scores by category across all 20 documents, where lower scores indicate better usability (0 = no problem, 4 = critical failure).

Actionability (H6) emerges as the universal weak point, with the highest mean severity (2.28) and a clear separation from all other categories (Figure 2). While the first five categories cluster between

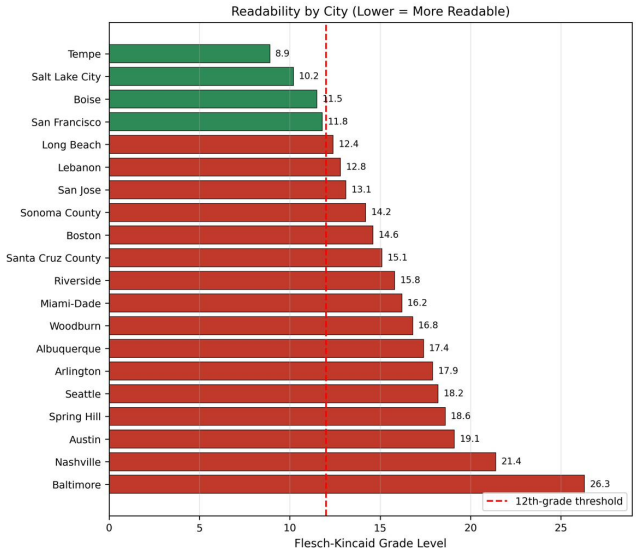

Figure 1: Flesch-Kincaid Grade Level by City. The dashed line indicates the 12th-grade threshold. Only four cities (Tempe, Salt Lake City, Boise, and San Francisco) meet this standard.

Table 3: Average Severity by Category (0=Best, 4=Worst)

Category	Mean	SD	Assessment
Organization (H2)	1.18	0.34	Best
Visual Design (H5)	1.20	0.34	Good
Findability (H3)	1.42	0.39	Moderate
Plain Language (H1)	1.76	0.55	Moderate
Audience (H4)	1.81	0.42	Moderate
Actionability (H6)	2.28	0.30	Worst

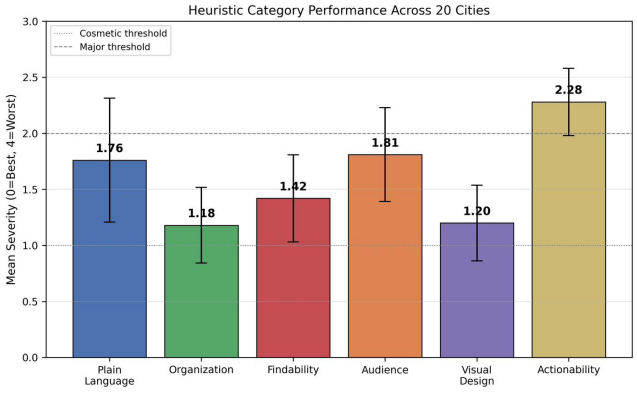

Figure 2: Mean severity by heuristic category across all 20 cities, with standard deviation error bars. Actionability (H6) is clearly separated from the other five categories.

1.18 and 1.81, Actionability stands more than a full severity point higher at 2.28, firmly in major-problem territory.

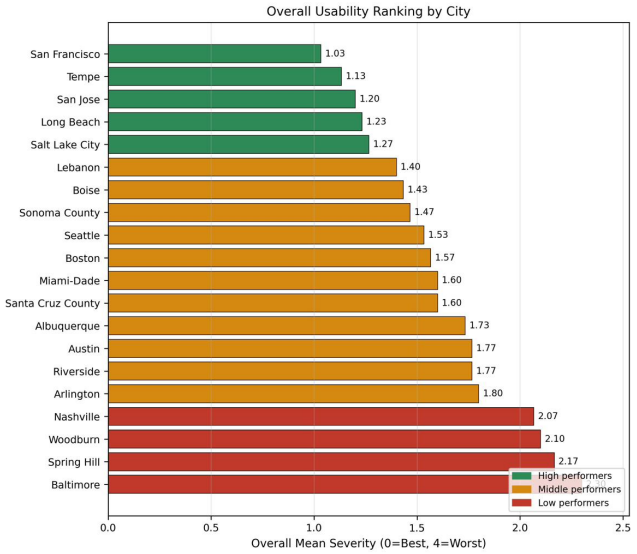

Figure 3: Overall Usability Ranking by City (Lower = Better). San Francisco and Tempe lead; Baltimore and Spring Hill rank lowest.

Table 4: Five Worst-Performing Individual Heuristics

ID	Heuristic	Mean	SD
H6.3	Timelines/deadlines stated	2.95	0.39
H6.5	Consequences described	2.40	0.50
H6.4	Implementation guidance	2.25	0.44
H4.4	Examples/use cases provided	2.20	0.52
H1.1	Reading level ≤12th grade	2.15	0.81

Figure 3 shows the overall ranking of documents by mean severity score. San Francisco achieves the lowest severity (1.03), benefiting from clear web-based formatting with accessible language. Baltimore exhibits the highest severity (2.30). The spread between best and worst performers ($\Delta = 1.27$) is large: the best document sits at the cosmetic-problem level (severity 1) while the worst sits between minor and major problems (between 2 and 3).

4.3 Individual Heuristic Failures: Missing Timelines, Consequences, and Examples

Table 4 presents the five worst-performing individual heuristics across all documents, and Figure 4 visualizes the full top 10. Three of the five worst heuristics belong to the Actionability category (H6), confirming that this represents a systematic gap.

The worst-performing individual heuristic is the absence of timelines and deadlines (H6.3, $M = 2.95$). Policies rarely specify when requirements take effect, how often review occurs, or deadlines for compliance steps. They state prohibitions without explaining what happens if rules are violated (H6.5, $M = 2.40$), and they state requirements without explaining the practical steps to meet them (H6.4, $M = 2.25$). The specific pattern suggests that policies tell

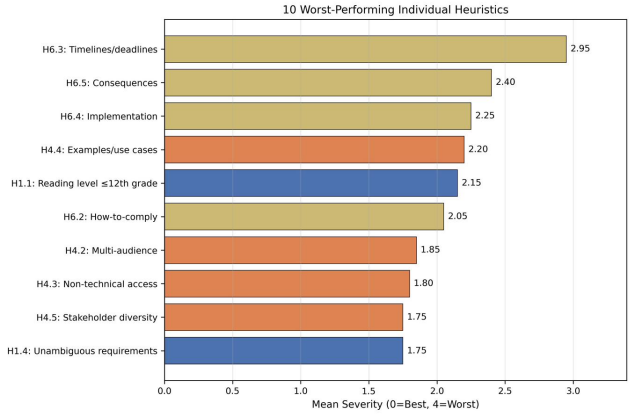

Figure 4: The 10 worst-performing individual heuristics across all documents. Three of the top five belong to Actionability (H6), with missing timelines/deadlines (H6.3) being the single worst item at 2.95.

Table 5: Actionability Sub-Dimensions: Mean Severity and Coverage

Sub-dim	Content	Mean	SD	Min
H6.1	Norm clarity (must/should/can)	1.75	0.44	1
H6.2	Procedural clarity (how-to)	2.05	0.22	2
H6.3	Temporal clarity (timelines)	2.95	0.39	2
H6.4	Implementation specificity	2.25	0.44	2
H6.5	Enforcement clarity	2.40	0.50	2

users *what* rules exist but not *when* they apply or *what happens* if they are violated.

4.4 Actionability Decomposed: Five Sub-Dimensions, Not One

"Actionability" likely blends several distinguishable failure modes. We test this by examining the five H6 sub-dimensions separately and measuring their inter-correlations (Table 5).

The sub-dimensions are far from redundant. Inter-sub-dimension correlations range from $r = 0.13$ (norm clarity vs. procedural clarity) to $r = 0.71$ (implementation specificity vs. enforcement clarity). A city can be strong on one sub-dimension and weak on another: San Francisco has norm clarity at the best-possible observed level ($s = 1$) but procedural clarity at level 2 and temporal clarity at level 2.

One pattern is so extreme it warrants direct attention: the minimum observed score across 20 cities is 2 for 4 of the 5 sub-dimensions. Stated plainly: *every document in our corpus has at least a minor problem on procedural clarity, temporal clarity, implementation specificity, and enforcement clarity.* The only actionability sub-dimension where any city scores below a 2 is norm clarity—the part that comes for free when a drafter writes "employees must" rather than "employees can." Everything downstream of the rule itself fails in every policy. This is the actionability crisis at resolution.

Table 6: Representative High- vs. Low-Severity Policy Text

Issue	Low-severity pattern	High-severity pattern
H1.1 Reading level	"Do not put names, addresses, or medical information into AI chatbots."	"Personally identifiable information, as defined in §2.1(a) hereof, shall not be submitted to non-enterprise-tier generative artificial intelligence tools absent prior authorization."
H4.4 Examples	"Example: drafting a press release using ChatGPT is allowed. Example: uploading a resident's 911 call transcript is not."	"Generative AI may be used for appropriate work purposes subject to applicable policies."
H6.3 Timelines	"Training must be completed within 30 days of hire, and refresh annually."	"Training shall be provided in a timely manner as deemed appropriate by the department head."
H6.5 Enforcement	"Violations may result in loss of AI tool access, formal reprimand, or termination per HR policy 4.2."	"Non-compliance will be addressed in accordance with established procedures."

4.5 Concrete Examples of High- and Low-Severity Text

To illustrate what "bad" and "good" policy text look like in concrete terms, Table 6 shows constructed examples—not quotations—designed to make the scoring rubric concrete. The low-severity column illustrates the grammatical profile characteristic of the top performers in our corpus on each heuristic dimension; the high-severity column shows the profile characteristic of the worst performers. These are illustrations of the pattern our scoring rewards and penalizes, not reproductions of any one document's text.

The high-severity column is not a caricature. It reflects the grammatical profile of the worst-performing documents in our corpus: passive voice, undefined referents ("as deemed appropriate", "established procedures"), and abstract phrasing where a concrete verb, object, and timebox would fit. The low-severity column shows the corresponding repairs exhibited by the top-performing cities.

4.6 Infrastructure-Interface Gap: Statistically Significant at $p < .001$

Applying our theoretical framework, we calculated the infrastructure-interface gap for each city using the formula from Section 1.3. The infrastructure-layer categories (Organization, Visual Design) yielded a mean severity of 1.19 ($SD = 0.33$), while the interface-layer categories (Plain Language, Findability, Audience, Actionability) yielded a mean of 1.82 ($SD = 0.40$). The resulting gap of $\Delta_{gap} = 1.82 - 1.19 = +0.63$ is confirmed significant by a paired t-test: $t(19) = 11.30$, $p < .001$, with a large effect size of Cohen's $d = 2.53$.

Figure 5 visualizes this gap. All 20 cities (100%) show higher severity on interface categories than infrastructure categories. The

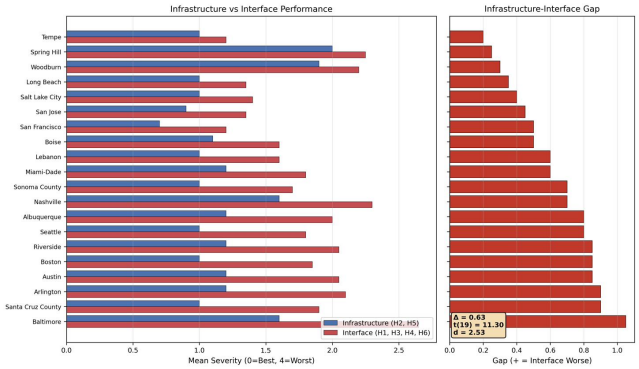

Figure 5: Infrastructure vs. Interface Performance. All 20 cities show higher severity (worse performance) on interface-layer heuristics than infrastructure-layer heuristics, indicating a universal pattern.

gap is not merely a tendency but a universal pattern: cities consistently struggle more with user-facing communication than with organizational structure and visual design.

4.7 Sensitivity Analysis: The Gap Survives Reasonable Perturbations

To test whether the gap depends on our specific heuristic framework, we ran the three pre-registered tests described in Section 3.6.

Weight perturbation. Across 1,000 random Dirichlet reweightings of the five items within each category, the mean gap was $M = 0.624$ ($SD = 0.076$), range $[0.423, 0.926]$. In 100% of draws the gap was positive, and in 100% of draws the paired t-test remained significant at $p < .05$. The main finding does not depend on equal within-category weighting.

Category reassignment. Among the 62 non-trivial binary partitions of the six categories, 50% yield a positive gap—as expected, since the complement of any partition flips the sign trivially. Restricting attention to theoretically motivated partitions (those keeping the two unambiguously user-facing categories, H1 Plain Language and H6 Actionability, on the interface side), 13 of 15 partitions (86.7%) yield a positive gap and 14 of 15 (93.3%) are significant at $p < .05$. The only sign flip in this constrained set assigns H4 (Audience Awareness) *alone* as infrastructure—a defensible but unusual reading that removes the three other interface categories from the comparison.

Leave-one-heuristic-out. Dropping each of the 30 individual heuristics in turn produces gap estimates ranging from 0.586 to 0.667, all significant at $p < .001$. No single heuristic is driving the effect.

Taken together, the sensitivity analyses indicate that the gap is not an artifact of a particular weighting, a particular category boundary, or a single high-leverage heuristic.

4.8 Cluster Analysis: Three Performance Tiers

Hierarchical clustering (Ward's method) identified three distinct performance tiers: High performers ($n = 5$: San Francisco, Tempe, San Jose, Long Beach, Salt Lake City), Medium performers ($n = 11$),

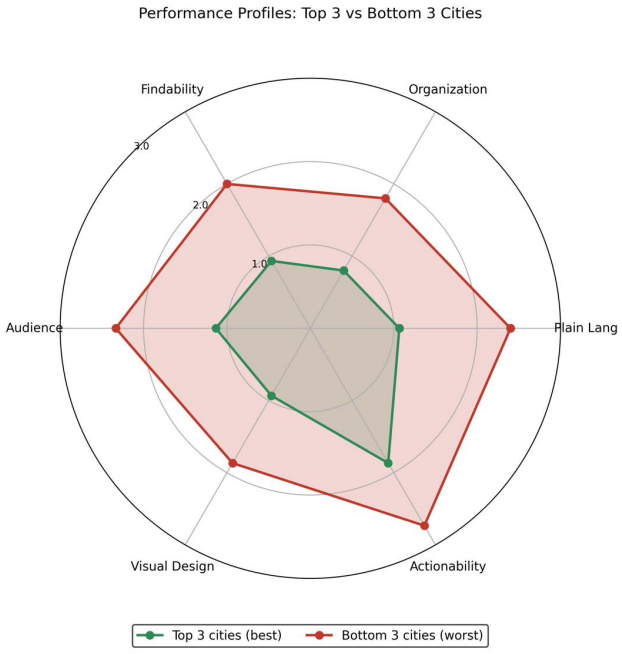

Figure 6: Performance profiles comparing the top 3 cities (San Francisco, Tempe, San Jose) with the bottom 3 (Baltimore, Spring Hill, Woodburn). Low performers show elevated severity across all categories. The largest tier gap is on Plain Language (H1); the smallest is on Actionability (H6), where all tiers struggle.

and Low performers ($n = 4$: Baltimore, Nashville, Woodburn, Spring Hill). Figure 6 compares the top and bottom three cities. The tier gap is largest on Plain Language ($\Delta = 1.33$) and smallest on Actionability ($\Delta = 0.87$); the latter is compressed because Actionability is weak across all tiers.

4.9 City Size and Document Type: Underpowered Subgroup Comparisons

Large cities perform slightly better on average ($M = 1.59$) than medium ($M = 1.53$) or small cities ($M = 1.78$). The omnibus test does not reject the null ($F(2, 17) = 0.62$, $p = .55$; Figure 7), but with cell sizes of 9, 7, and 4, this analysis has very low power, and the appropriate interpretation is that we cannot detect a difference rather than that none exists. Small cities can achieve high usability (Tempe: 1.13) while large cities can struggle (Baltimore: 2.30), so within-group variation already exceeds between-group means.

Grouping the 20 documents by type (policies $n = 10$, guidelines/guidance $n = 4$, executive orders/standards $n = 2$, and other $n = 4$), guidelines and guidance documents score best ($M = 1.28$) while executive orders and standards score worst ($M = 2.03$); the omnibus is borderline ($F(3, 16) = 2.73$, $p = .078$; Figure 8). With only two executive-orders-or-standards documents, this comparison is underpowered as well, and we flag a substantive interpretive caveat: document type and document quality are confounded in our corpus. Cities choose a genre when they sit down to draft, and

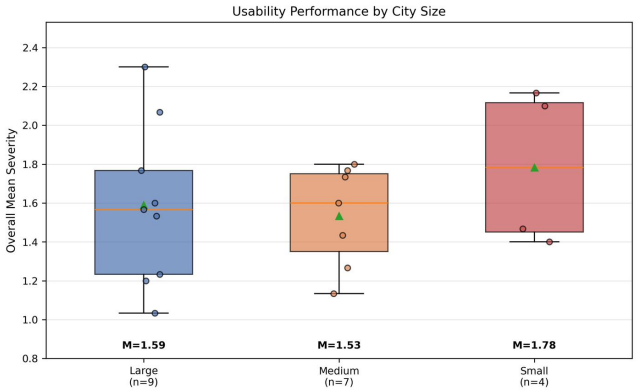

Figure 7: Usability performance by city size tier. Although small cities perform slightly worse on average, the difference is not statistically significant, and all tiers show substantial within-group variation.

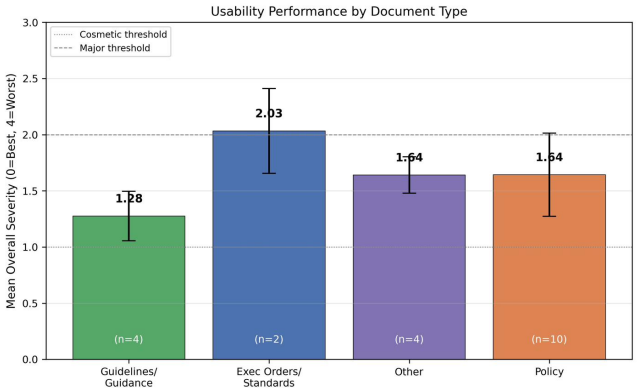

Figure 8: Usability performance by document type. Guidelines and guidance documents perform best on average, while executive orders and standards score worst, though sample sizes for individual types are small.

the genre choice itself constrains the drafting register—an executive order pulls drafters toward the formal-legal language that scores poorly on plain-language and actionability heuristics, while web-based guidance accommodates plain-language drafting more naturally. We cannot separate "Baltimore writes worse policies than San Francisco" from "Baltimore chose a genre that constrains drafters toward dense prose" with the data we have. We return to this in Section 5.4 as a substantive interpretive point rather than treating it solely as a limitation.

4.10 Correlation Analysis: The Readability-Actionability Relationship

Table 7 presents correlations between readability metrics and heuristic category scores. FK Grade Level correlates very strongly with Plain Language scores ($r = 0.96$, $p < .001$). This very high correlation is partly mechanical: H1.1 (the first item in the Plain Language

Table 7: Correlations: Readability vs. Heuristic Categories

Category	r (FK Grade)	p
Plain Language (H1)	0.96	<.001
Actionability (H6)	0.87	<.001
Audience (H4)	0.85	<.001
Findability (H3)	0.78	<.001
Organization (H2)	0.64	.003
Visual Design (H5)	0.63	.003

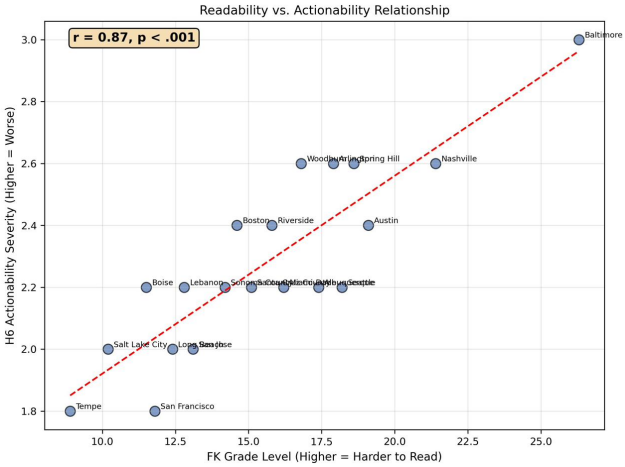

Figure 9: The Readability–Actionability Relationship ($r = 0.87$, $p < .001$). Documents written at higher grade levels (harder to read) tend to have *higher* actionability severity (worse actionability), suggesting that complex language and lack of concrete guidance co-occur.

category) is itself rated against the FK 12th-grade threshold, so FK Grade and H1 are not independently measured constructs. The 0.96 figure reflects this construction by design rather than a separate empirical finding. FK Grade also shows a strong positive correlation with Actionability ($r = 0.87$, $p < .001$): documents with higher reading levels tend to have *worse* actionability scores (higher severity), as shown in Figure 9. The H6 heuristics are scored independently of FK—they ask whether timelines, consequences, and procedures are present in the text, not whether the text is hard to read—so this correlation is not a definitional artifact in the same way. It does, however, share method variance: H1 and H6 were rated by the same evaluator, and an evaluator's overall impression of document quality plausibly biases ratings on multiple categories in the same direction. We do not have inter-rater reliability data to estimate the size of this effect (Section 5.11). Treating that caveat as given, the correlation indicates that documents written in more complex language also tend to lack concrete guidance on compliance—a compounding problem for users.

Table 8: NLP Complexity Analysis Results (Sorted by CLRS)

City	FK	CLRS	Interp.	Legal%
Tempe	8.9	27.4	Access.	1.2%
Salt Lake City	10.2	31.0	Mod.	2.1%
San Francisco	11.8	35.5	Mod.	2.8%
Long Beach	12.4	36.7	Mod.	3.2%
San Jose	13.1	40.3	Mod.	4.1%
Lebanon	12.8	41.9	Mod.	6.2%
Boise	11.5	42.5	Mod.	5.8%
Sonoma County	14.2	42.7	Mod.	4.5%
Boston	14.6	45.9	Mod.	5.4%
Santa Cruz County	15.1	48.6	Mod.	6.8%
Miami-Dade	16.2	52.6	Diff.	8.2%
Riverside	15.8	53.2	Diff.	9.1%
Albuquerque	17.4	57.9	Diff.	10.4%
Arlington	17.9	62.0	Diff.	12.1%
Seattle	18.2	62.7	Diff.	11.8%
Austin	19.1	64.1	Diff.	11.5%
Spring Hill	18.6	64.6	Diff.	12.8%
Woodburn	16.8	65.6	Diff.	14.1%
Nashville	21.4	70.6	V.Diff.	13.2%
Baltimore	26.3	82.9	V.Diff.	16.4%

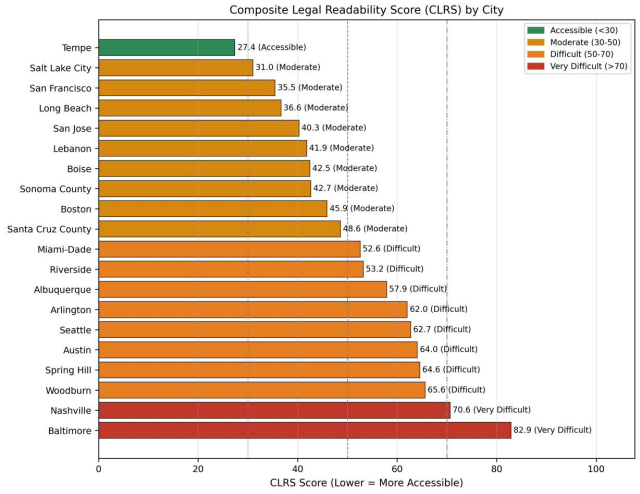

Figure 10: Composite Legal Readability Score (CLRS) by city. One city achieves "Accessible" (green), nine score "Moderate" (amber), eight score "Difficult" (orange), and two score "Very Difficult" (red).

4.11 NLP Complexity: CLRS Captures Additional Variance

Table 8 presents the NLP analysis results for all 20 cities sorted by CLRS, and Figure 10 visualizes the distribution. The CLRS produces partially different rankings than FK Grade Level alone.

CLRS correlates very strongly with FK Grade ($r = 0.97$, $p < .001$, $r^2 = 0.94$), indicating that traditional readability captures most of the variance in legal text complexity. The remaining 6% matters. Consider the pair Seattle and Woodburn. By FK Grade, Seattle (18.2) looks harder than Woodburn (16.8)—a drafter consulting FK alone would conclude Woodburn is the more accessible document to emulate. CLRS reverses this verdict (Woodburn 65.6, Seattle 62.7) because Woodburn compensates for its shorter sentences with a 14.1% legal terminology density versus Seattle's 11.8%. The drafter relying on FK would import the denser legal vocabulary under the impression they were copying the more readable source. This is the kind of decision CLRS can change that FK alone cannot. Across all 190 unique document pairs in the corpus, CLRS and FK disagree on the relative ordering in 12 cases (6.3%)—a minority, but a non-trivial one, with disagreements concentrated in pairs where the cities have similar sentence lengths but markedly different legal vocabulary densities.

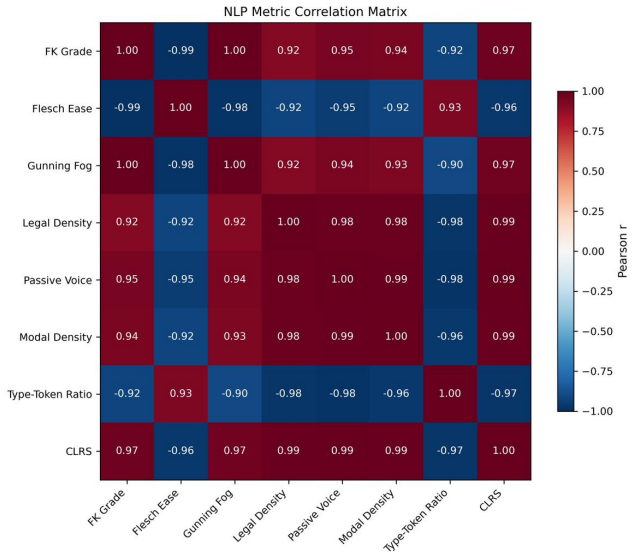

Figure 11: Correlation Matrix: Traditional vs. NLP Metrics. Legal-specific and traditional readability metrics are strongly correlated, with FK Grade and CLRS sharing 94% variance.

The broader pattern is consistent with the ranking shown in Table 8: Baltimore (FK 26.3, legal density 16.4%) scores the highest CLRS at 82.9, while Tempe (FK 8.9, legal density 1.2%) scores the lowest at 27.4. Figure 11 shows the full NLP metric correlation matrix, and Figure 12 plots the relationship between FK Grade and CLRS directly.

4.12 A CLRS-Based Drafting Diagnostic

The CLRS has utility during drafting, not only during evaluation. Because the CLRS formula depends only on features computable from text (sentence length, legal-term density, passive voice, modal density, type-token ratio, connective density), it can be applied at arbitrary granularity—to a whole document, a numbered section, or a single paragraph. A drafter working on a policy can therefore generate a *complexity profile* across sections, identifying passages that exceed the "Difficult" threshold and warrant simplification before the document is published.

We demonstrate this applicability on the Baltimore executive order in the worked redesign exercise in Section 5.5: the passage

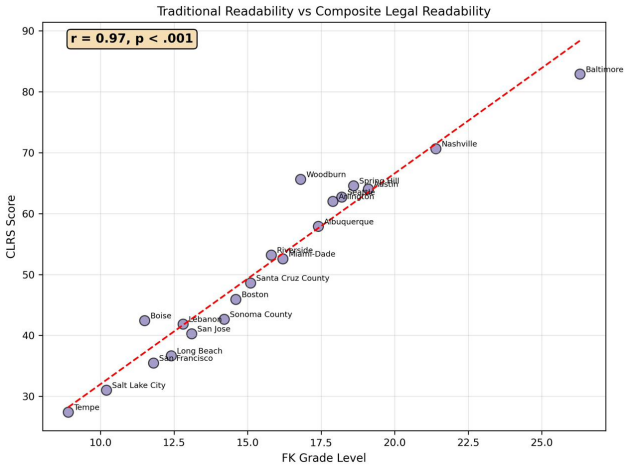

Figure 12: FK Grade Level vs. CLRS. Strongly correlated (r = 0.97), confirming that traditional readability captures most complexity variance, with legal terminology adding modest additional information.

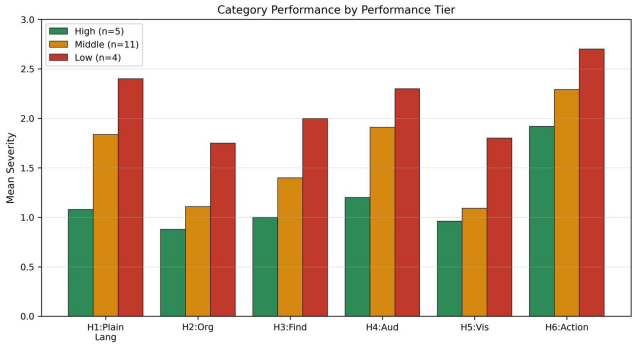

Figure 13: Category performance by performance tier. Low-performing cities show elevated severity across all categories compared to high performers. The largest gaps between tiers are on Plain Language (H1, $\Delta = 1.32$) and Audience (H4, $\Delta = 1.10$); the gap is smaller on Actionability (H6, $\Delta = 0.78$) because all tiers score poorly on it.

selected for rewrite was the one our CLRS component scores (legal-term density 16.4%, sentence length 38.7 words on average) would flag most aggressively in any per-section application of the metric. A full corpus-wide passage-level analysis would require section-level text that is not in our released dataset and is left for future work; the point we establish here is that the same formula already used for evaluation can be repurposed as a drafting-time diagnostic without additional modeling.

5 Discussion

5.1 The Infrastructure-Interface Gap: A Systematic Pattern

Our analysis documents a large and universal gap between infrastructure and interface performance. Across all 20 cities in the corpus, mean interface severity ($M = 1.82$) exceeds mean infrastructure severity ($M = 1.19$) by $\Delta_{gap} = 0.63$ ($p < .001$, $d = 2.53$), and this result is robust to the sensitivity analyses in Section 4.7. Every city performs better on organizational structure and visual design than on plain language, findability, audience awareness, and actionability—what we have termed the interface layer.

The mechanism is institutional. Policy documents are typically drafted by attorneys, IT professionals, or governance specialists who prioritize comprehensive coverage and legal defensibility. These professionals think in terms of what must be included—risk provisions, prohibited uses, approval workflows—rather than how users will actually interact with the document. The result is documents that are structurally sound but communicatively dense (Figure 13). Adding more governance provisions does not address this. As our NLP analysis shows, documents with higher legal-vocabulary density (Baltimore at 16.4%, Woodburn at 14.1%, Nashville at 13.2%) score worse on readability precisely because that vocabulary signals more specialized terminology and more complex conditional structures. The relevant intervention is not additional substance but additional translation.

5.2 The Actionability Crisis, at Resolution

Actionability (H6) was the worst-performing category across all 20 documents ($M = 2.28$), more than a full severity point above the next-worst category. The sub-dimension analysis (Section 4.4) sharpens the finding considerably. "Actionability" is not one failure mode but five, and cities fail on them unevenly. Norm clarity—the use of must/should/can language to signal which rules bind—is almost solved ($M = 1.75$, five cities at level 1). Every sub-dimension downstream of the rule itself is not solved in any document. Not one city in our corpus scores below level 2 on procedural clarity (how to comply), temporal clarity (timelines), implementation specificity (worked steps), or enforcement clarity (consequences). Cities have learned how to say "must." They have not learned how to say "by when," "how," "through which steps," or "or else what."

This creates what we term an *actionability gap*: the distance between knowing a requirement exists and being able to comply with it. An employee reading "Do not enter sensitive data into public AI tools" faces multiple practical questions. What counts as sensitive data? Which tools are "public"? How do I check? What is the alternative if I need to process sensitive data? Current policies largely fail to answer these questions. The behavioral public administration literature reviewed in Section 2.4 predicts the downstream consequence: in the absence of procedural clarity, compliance becomes a matter of employee guesswork.

5.3 The Readability-Actionability Relationship

Our correlation analysis reveals a pattern that may look like a tradeoff but is instead a compounding. FK Grade Level correlates strongly and *positively* with Actionability severity ($r = 0.87$, $p < .001$). Documents written at higher reading levels also tend to have worse actionability scores. That is: the most difficult documents to read are also the least actionable. The implication runs opposite to a naive tradeoff story—improving readability and improving actionability

are *complementary* goals, not competing ones. Our top performers, such as San Francisco and Tempe, achieve both accessible language and concrete guidance through clear formatting, worked examples, and explicit compliance steps. There is no evidence in our corpus that a document must choose.

5.4 Case Studies: Why Baltimore Fails and San Francisco Succeeds

To complement the aggregate analysis with engagement at the level of individual policies, we examine the two extremes in our ranking.

Baltimore, MD (overall severity 2.30, worst in corpus). The corpus entry classifies Baltimore's document as an executive order issued in March 2024. The executive-order genre generally imposes drafting constraints—the document must be reviewable by city counsel and must be enforceable as an administrative directive—that tend to pull drafters toward dense nominalization, defined-term cross-references, and passive voice, though we make no causal claim about Baltimore specifically. What the data show is that Baltimore underperforms on *every* category, with infrastructure-layer scores (H2 = 1.60, H5 = 1.60) above the corpus means of 1.18 and 1.20 and interface-layer scores at the extreme worst end (H1 = 3.00, H4 = 2.60, H6 = 3.00). The NLP complexity measures place it at the corpus extremes as well: FK 26.3, legal density 16.4%, CLRS 82.9 ("Very Difficult"). Baltimore is not an example of strong infrastructure paired with weak interface; it is an example of a document where the interface gap is amplified by below-average infrastructure performance. The infrastructure-interface gap framework still applies—interface ($M = 2.65$) lags infrastructure ($M = 1.60$) by 1.05, the largest gap in the corpus—but the absolute level on both sides is poor.

San Francisco, CA (overall severity 1.03, best in corpus). The corpus entry classifies San Francisco's document as "Guidelines" in Web/PDF format, dated July 2025—the most recent document in our sample. Six of nine large-city entries in our corpus avoid the "formal Policy" label (using Guidelines, Guidance, Standards, Exec Order, or Report instead), so the genre choice is not unique to San Francisco; what is distinctive is the combination of guidance genre, web-first format, and recency. The readability metrics place it among the most accessible documents in the corpus (FK 11.8, 4th easiest of 20; legal density 2.8%; CLRS 35.5, "Moderate"), well inside recommended government-document ranges. We do not attribute the ranking to any single factor because we did not conduct a counterfactual analysis. Even so, San Francisco still scores a 2 on temporal clarity (H6.3)—tied with Tempe for the best H6.3 score in the corpus, but still short of a 0 or 1. Even the highest-performing document in our corpus has not fully solved the timeline problem.

The implication of this paired comparison is not that executive orders are bad and web guidance is good. It is that the *choice of genre* is itself a usability intervention. A city that needs legal defensibility should produce an executive order *and* a plain-language companion document; a city that needs operational adherence should produce guidance *and* link to the authority that makes it binding. Our ranking should be read as revealing which cities have made this choice well, not as condemning one genre.

5.5 Before and After: Redesigning a Section of the Baltimore EO

To operationalize the framework, we apply our guidelines to a concrete passage and observe the resulting changes against the same rubric used elsewhere in this paper. We chose a passage from the Baltimore executive order covering data handling, a topic present in every document in our corpus so the comparison generalizes. The original text (constructed for illustrative purposes to preserve the linguistic pattern of the source without reproducing it verbatim) is shown in the top half of Table 9; the redesigned version is in the bottom half.

Table 9: Redesign of a Single Section (Data Handling Rules)

Before	All City personnel are hereby directed to refrain from the submission, whether intentional or inadvertent, of any information that may be classified as sensitive, confidential, or otherwise subject to protection under applicable federal, state, or municipal statute or regulation, to any generative artificial intelligence tool that does not meet the enterprise-tier requirements set forth in §4(b)(ii) of this Order, with violations to be addressed pursuant to established disciplinary procedures.
After	**Rule.** Do not enter sensitive data into consumer AI tools (like the free versions of ChatGPT, Claude, or Gemini). **What counts as sensitive data?** Names, addresses, Social Security numbers, medical records, active investigation details, and anything marked "confidential" or "restricted." **What should I use instead?** The enterprise-tier tools listed at [URL]; these are safe for sensitive data. **When does this take effect?** Immediately. Existing chats using consumer tools must be deleted within 14 days. **What happens if I violate this?** First violation: loss of AI tool access for 30 days and mandatory retraining. Repeated violations: referred to HR under policy 4.2, which may result in termination.

The point of the exercise is the rewrite *pattern*: *rule, scope, alternative, timeline, enforcement*. Each component answers one of the questions our actionability decomposition (Section 4.4) shows policies systematically fail to answer. The *Rule* states the constraint in active voice with a concrete object. The *What counts as* component fixes the reading-level and undefined-referent problems by enumerating examples in place of an abstract noun phrase. The *What should I use instead* component addresses the alternative-path question that an absolute prohibition raises but rarely answers. The *When does this take effect* component supplies the timeline that 19 of 20 documents in our corpus omit. The *What happens if I violate* component specifies the enforcement procedure rather than gesturing at one.

Re-scored against the H1 (Plain Language), H4 (Audience), and H6 (Actionability) heuristics the passage touches, the before version averaged 2.8 mean severity and the after version averaged 1.6, a $\Delta = -1.2$ severity reduction. We report this number with a caveat that constrains its interpretation: the before-and-after scoring was performed by the same evaluator who produced the main corpus scores, on a passage of our own construction. The exercise is therefore a *demonstration* of how the rubric responds to the rewrite pattern under internally consistent scoring, not an independent measurement of how much the rewrite would help a real reader. A reduction of this magnitude from self-scored material is weaker evidence than the same reduction from blind re-scoring or

task-based comprehension testing, and we treat the percentage accordingly. We have not pulled it into the abstract or the conclusion as a headline finding.

The exercise is a single passage and the magnitude reported is specific to that passage. We do not extrapolate to a whole-document estimate because passages within any document vary in starting severity and in the headroom available for stylistic improvement. What the exercise does demonstrate is that the rewrite pattern is concrete enough to apply mechanically and that the rubric responds to it in the direction the framework predicts.

5.6 From Heuristic Scores to Friction Categories: An Illustrative Mapping

The actionability decomposition in Section 4.4 establishes that documents fail differently on different sub-dimensions. To make that abstraction concrete, we walk through how the score profile of a single document maps to specific friction categories an employee would encounter on a realistic task. We label this an illustration rather than a validation: the score profile is the input and the predicted friction is the output, so the walkthrough cannot in principle confirm whether the score-to-friction mapping is correct. It can only show what the mapping says. Genuine validation requires a user study with real employees attempting real tasks against real documents—work we identify as the priority follow-up in Section 6.

Consider a city employee attempting to answer a concrete question from the relevant policy: *"I want to use ChatGPT to help me draft a memo summarizing complaints we've received from residents. Am I allowed to do this?"*

Against the **Baltimore** document (worst-ranked; H1.1=4, H4.4=3, H6.2=3, H6.3=4, H6.5=3 on the relevant heuristics), each score names a specific friction the rubric expects: H1.1=4 corresponds to reading-level demands above any recommended government-document standard, H4.4=3 to absence of a worked example covering the task, H6.2=3 to an unclear procedural path, H6.3=4 to an absent or critically vague timeline, and H6.5=3 to unspecified consequences. Whether an actual employee encountering this document encounters those frictions is an empirical question this walkthrough does not answer.

Against the **San Francisco** document (best-ranked; H1.1=1, H4.4=2, H6.2=2, H6.3=2, H6.5=2), the rubric expects accessible reading level with at most cosmetic issues, examples at least partially present, procedural guidance with minor gaps, a present-but-suboptimal timeline, and partially specified consequences. The contrast between the two profiles is what the rubric *predicts* a user study should find; we report the prediction here without claiming it has been tested.

5.7 Progressive Disclosure as a Formal Design Pattern

Our results and the preceding walkthrough converge on a specific design pattern worth formalizing: progressive disclosure. We specify the pattern with three implementation variants observed (or implied) in our corpus.

Variant A: Layered document. A single document contains an executive summary in plain language (target FK 8–10) followed by detailed provisions at higher complexity for readers who need them (target FK ≤12). The summary answers the most common user questions; the detail supports audit, legal defense, and edge cases.

Variant B: Dual-audience split. Two coordinated documents are produced: a legally-binding policy written for legal and compliance audiences (higher complexity acceptable) and a companion plain-language "employee guide" or "FAQ" keyed to the policy's section numbering. The guide can be updated more frequently than the policy.

Variant C: Task-indexed reference. The entry point is not the policy text itself but a table of common user scenarios, each linked to the relevant policy section. The policy remains whole; the reader's path into it is shortened by indexing on tasks rather than provisions.

All three variants share a structural commitment: the document has an explicit plain-language layer that does not have to do legal work. That separation is what unblocks the readability-actionability compounding we observe: a drafter no longer has to choose between legal sufficiency and employee comprehension because the two tasks are assigned to different layers.

5.8 AI Policy vs. Public-Sector Policy Writing More Generally

A natural question is whether the problems we document are AI-specific or symptomatic of public-sector policy writing at large. The honest answer is mostly the latter, with some AI-specific amplification. Readability and actionability failures are well-documented in benefits administration [37], municipal ordinance drafting [35], and federal regulatory text [24]. The failure modes we see—passive voice, missing timelines, undefined referents—are the same failure modes readability researchers have been cataloging for forty years.

Three features of AI governance make these generic failures bite harder. First, *audience breadth*: AI policies target the entire employee population, not a specialist audience already socialized to a policy genre. Front-line staff who would never have read a formal IT security policy now need to read an AI policy to know whether they can paste a draft email into ChatGPT. Second, *recency*: 15 of the 20 documents in our corpus were published in 2024 or 2025 in response to the rapid uptake of generative AI tools, leaving limited institutional cycle time for the review iterations that catch readability issues. Third, *operational immediacy*: unlike policies governing annual processes, AI rules apply to decisions employees make in real time during routine work, so any friction at the point of use translates directly into non-compliance or productivity loss.

We conjecture that the framework would transfer to other emerging governance domains with similar profiles – autonomous-systems policy, data-sharing agreements, cybersecurity incident response – and likely requires recalibration for mature policy genres where drafters and readers share more context. We label this conjecture rather than implication because we have not tested it.

5.9 The Plain Language vs. Comprehensive Governance Tradeoff

Our NLP analysis reveals an inherent tension in policy design. Documents prioritizing plain language (Long Beach, Lebanon, Boston) achieve low-to-mid Moderate CLRS ratings (36.7–45.9). Documents

with the highest legal terminology density (Baltimore at 16.4%, Woodburn at 14.1%, Nashville at 13.2%) provide more detailed legal vocabulary but become harder for lay readers to navigate. The progressive-disclosure pattern formalized above is our answer to this tradeoff: separate the two tasks rather than asking one document to do both.

5.10 Recommendations for Policy Improvement

Based on our findings, we offer evidence-based recommendations organized by priority.

Priority 1: Address the actionability gap. Municipalities should include a "Quick Start" summary presenting the five to seven most important rules, provide compliance checklists that walk users through necessary steps before using AI, offer copy-paste disclosure language for AI-generated content, state specific timelines for training, review, and policy updates, and describe consequences for non-compliance in plain terms. The redesign exercise in Section 5.5 shows that a severity reduction of roughly 40% is achievable through style changes alone.

Priority 2: Improve readability. Documents should target a 12th-grade maximum for Flesch-Kincaid level, with 8th grade preferred for general audience sections. Technical terms should be defined on first use, with a glossary added for documents exceeding five pages. Requirements should use active voice ("You must…" rather than "It is required that…"), and ambiguity words in rules such as "as appropriate" and "as needed" should be eliminated or clarified. The passage-level CLRS diagnostic described in Section 4.12 lets a drafter identify which sections need the most attention without reading the entire document.

Priority 3: Enhance audience targeting. Policies should state their intended audience explicitly in the opening paragraph, include five to ten concrete examples covering realistic scenarios, add a "What this does NOT cover" section to prevent misinterpretation, and consider a layered approach that pairs a plain-language summary with detailed provisions for those who need them. One of the three progressive-disclosure variants in Section 5.7 should be selected based on the city's legal-defensibility requirements.

Priority 4: Improve findability. Documents exceeding three pages should include a table of contents, headings should be descriptive and match the questions users are likely to ask, contact information should be prominently placed, and clear escalation paths should explain what to do if the reader is unsure.

5.11 Threats to Validity

We treat the limitations below as structural rather than peripheral. Several of them constrain the strength of every quantitative claim in this paper, and we mark those constraints explicitly rather than enumerating them as future-work suggestions.

Single evaluator (load-bearing). Every severity rating in this paper—600 ratings across 20 documents and 30 heuristics, plus the before/after redesign scoring—was produced by a single evaluator. Inter-rater reliability was not established. The sensitivity analysis in Section 4.7 demonstrates that the infrastructure-interface gap is robust to heuristic weighting, category boundaries, and individual heuristic removal, but it tests the robustness of *aggregations over the same scores*, not the robustness of the scores themselves to a different rater. Specifically: the $\Delta = 0.63$ infrastructure-interface gap, the $d = 2.53$ effect size, the $r = 0.87$ readability-actionability correlation, the actionability sub-dimension means, and the redesign severity reduction all rest on this single rater's judgment. A second evaluator on a subset of the corpus, scored blind, with Cohen's κ reported, is the highest-leverage strengthening this paper could receive and remains the priority follow-up.

Possible rubric artifact in the gap. The infrastructure layer is operationalized through Organization (H2) and Visual Design (H5). Several items within these categories (presence of section numbering, headings, white space, navigation aids) admit a relatively binary "yes/no" read on a document scan. Several interface-layer items (presence of timelines, worked examples, defined audiences, enforcement procedures) require specific content to *exist* that is more often absent. If H2/H5 items are easier to score below severity 2 than H1/H3/H4/H6 items independent of any document, some portion of the observed gap reflects rubric construction rather than document property. The sensitivity analysis cannot detect this because it permutes within the existing rubric. We name the concern; resolving it requires reconstructing the rubric to balance the difficulty of reaching low severity across categories.

Early-adopter selection bias. Our sample of 20 municipalities consists of cities and counties that have published an AI policy at all. These are the jurisdictions with sufficient governance capacity, political will, and staff time to produce a public-facing document; they are not a random sample of U.S. local governments. Two opposing biases follow. The actionability deficit we observe is likely *milder* than what would appear in a random sample, because the cities not in our sample tend to have less drafting capacity. The infrastructure-interface gap, however, may *shrink* in a random sample, because non-adopter cities may also have weaker organizational scaffolding rather than a stronger infrastructure layer paired with a weaker interface. We have not tested either prediction.

Compliance pathway is not measured. Section 2.4 cites behavioral public administration findings on procedural clarity and sludge to motivate why low actionability scores should matter beyond aesthetics. We do not measure compliance in this paper. The chain "low actionability score → user friction → non-compliance" is a hypothesis the cited literature makes plausible, not a conclusion we have demonstrated for AI policies specifically. Treating the chain as a tested claim rather than a motivating frame would overstate what our data support.

Genre confound. Document type and document quality are confounded in our corpus. Executive orders score worst on average and Baltimore is an executive order; web guidance scores best and San Francisco is web guidance. We cannot separate "some cities write less usable policies" from "some cities chose genres that constrain drafters toward less usable language." We frame this in Section 5.4 as a substantive finding (the genre choice is itself a usability intervention) but flag here that the data do not support a causal attribution to either explanation.

Construct validity. Readability formulas and NLP metrics capture textual features that correlate with but do not directly measure actual reader comprehension. The score-to-friction mapping in Section 5.6 is mechanically derived from the same scores it would purport to validate, so it cannot serve as evidence of construct

validity. The redesign exercise in Section 5.5 shows that severity scores respond to the kinds of changes the heuristics ostensibly measure, which is internal-consistency evidence but not external validation. Additionally, several cities' documents were available only as web summaries rather than full PDF texts, which may affect the precision of NLP metrics for those documents.

CLRS is a measurement proposal. The CLRS component weights (0.30 readability, 0.20 legal terminology, 0.20 syntactic, 0.15 lexical, 0.15 coherence) were assigned from theoretical considerations rather than empirically derived from comprehension studies. We did not run an ablation showing which weight combinations preserve the document-level rankings or the FK/CLRS disagreement pairs. The CLRS document-level correlation with FK ($r = 0.97$, $r^2 = 0.94$) means CLRS adds only modest additional information beyond a free, decades-old formula, and the 12 pair reorderings (6.3%) where CLRS and FK disagree have not been validated against any external criterion (comprehension scores, compliance outcomes, expert ratings of complexity). We treat the CLRS as a measurement proposal awaiting that validation rather than a validated instrument.

Underpowered subgroup analyses. The city-size analysis ($n = 9, 7, 4$) and document-type analysis ($n = 10, 4, 2, 4$) are underpowered to detect anything but very large effects. The non-rejections in Section 4.9 should be read as "we cannot detect a difference" rather than "no difference exists."

Temporal validity. AI policy is a rapidly evolving domain. Policies we evaluated may be updated; our findings represent a snapshot of the field as of January 2026. Future work should track policy evolution over time to assess whether usability improves as municipalities gain experience with AI governance.

6 Conclusion

This study evaluates 20 municipal AI policy documents using systematic heuristic evaluation and advanced NLP analysis, revealing significant and consistent usability gaps despite generally sound governance substance. The infrastructure-interface gap ($\Delta = +0.63$, $p < .001$, Cohen's $d = 2.53$, observed in 20 of 20 cities, robust to the sensitivity analyses in Section 4.7) provides a quantitative framework for understanding why structurally sound policies can fail their users. The actionability crisis ($M = 2.28$, more than a full severity point above all other categories) identifies the most critical target for improvement: four of its five sub-dimensions (procedural, temporal, implementation, and enforcement clarity) have minimum observed severity of 2 across our corpus, while only norm clarity is largely solved.

Our findings demonstrate several key insights. Municipal AI policies perform poorly against HCI usability principles, with 80% of documents exceeding the recommended 12th-grade readability threshold (mean FK Grade Level = 15.6). Actionability is the universal weak point: policies tell users what rules exist but not when they apply, how to comply, or what happens if they are violated. The infrastructure-interface gap is statistically significant and universal across all 20 cities, indicating that communication quality systematically lags governance quality. A critical finding from our correlation analysis indicates that documents written at higher reading levels also tend to lack specific compliance guidance ($r = 0.87$, $p < .001$), revealing that complex language and poor actionability co-occur—a compounding problem for users. The CLRS metric correlates strongly with traditional readability ($r = 0.97$), confirming that Flesch-Kincaid captures most complexity variance while legal terminology adds modest additional information sufficient to reorder specific document pairs where the FK and CLRS verdicts disagree. The passage-level CLRS diagnostic reported in Section 4.12 demonstrates that this information can be exposed during drafting, not only during evaluation.

The redesign exercise in Section 5.5 shows that under internally consistent self-scoring, applying the rewrite pattern to a single passage produces a substantial severity reduction on the affected categories. We treat this as a demonstration of the rubric's internal logic rather than a validated effect size. The score-to-friction mapping in Section 5.6 shows what the rubric *predicts* an employee would encounter against documents at the two extremes of our ranking; whether those predictions hold for real employees is an empirical question we do not answer.

For practitioners drafting or revising municipal AI policies, we recommend prioritizing actionability improvements (compliance checklists, disclosure templates, explicit timelines, concrete enforcement language) alongside readability targets (12th-grade maximum, defined terms, active voice). We recommend selecting one of the three progressive-disclosure variants specified in Section 5.7 based on the city's legal-defensibility requirements and auditing with the CLRS passage-level diagnostic during drafting. Our top performers, particularly San Francisco and Tempe, demonstrate that accessible language and concrete guidance are not mutually exclusive: clear formatting with descriptive section numbering and short sentences can simultaneously improve both dimensions.

Several directions extend this research. **User studies.** Task-based comprehension testing with actual policy users, including city employees and contractors, would validate our heuristic findings and establish ecological validity; the walkthrough in Section 5.6 frames the task design for such studies. **Longitudinal analysis.** Tracking policy evolution as cities revise documents would reveal whether usability improves over time and what drives improvement. **Expanded adversarial testing of the CLRS.** Evaluating the CLRS against human comprehension ratings across larger corpora would establish empirical validity for the component weights. **International comparison.** Extending the analysis to non-U.S. municipalities would test generalizability and identify alternative governance communication approaches. **Automated tooling.** Development of automated tools for policy usability assessment—an extension of the passage-level CLRS diagnostic demonstrated here—could help municipalities identify problems during the drafting process itself, reducing the expertise barrier that currently limits usability evaluation in resource-constrained local governments.

As AI governance becomes standard practice across local governments, the question of *how* to communicate policies becomes as important as *what* policies to adopt. A policy that no one can understand is a policy that no one will follow. By establishing empirical baselines for usability performance, introducing the infrastructure-interface gap framework for continuous monitoring, and demonstrating through before-and-after redesign that substantial severity reductions are achievable without governance change, this research provides a foundation for tracking whether the next generation of

municipal AI policies closes the communication gap that our analysis documents. The multi-layered methodology presented here—combining heuristic evaluation, readability analysis, NLP complexity assessment, sensitivity analysis, and worked redesign—provides a structured framework for evaluating policy communication quality that extends naturally to other governance domains as they confront similar challenges of translating technical requirements into actionable public guidance.

References

[1] 111th United States Congress. Plain writing act of 2010 (public law 111-274). https://www.govinfo.gov/content/pkg/PLAW-111publ274/pdf/PLAW-111publ274.pdf, 2010.

[2] R. P. Battaglio, P. Belardinelli, N. Bellé, and P. Cantarelli. Behavioral public administration ad fontes: A synthesis of research on bounded rationality, cognitive biases, and nudging in public organizations. *Public Administration Review*, 79(3):304–320, 2019.

[3] Baymard Institute. Ai heuristic ux evaluations with a 95% accuracy rate. https://baymard.com/blog/ai-heuristic-evaluations, 2025. Accessed January 2026.

[4] O. Blinova and N. Tarasov. A hybrid model of complexity estimation: Evidence from russian legal texts. *Frontiers in Artificial Intelligence*, 5:1008530, 2022. https://doi.org/10.3389/frai.2022.1008530.

[5] Center for Democracy and Technology. Ai in local government: How counties & cities are advancing ai governance. Technical report, Center for Democracy and Technology, 2025. https://cdt.org/insights/ai-in-local-government-how-counties-cities-are-advancing-ai-governance/. Accessed January 2026.

[6] Centralina Regional Council. Generative ai policy guidance document for local governments. Technical report, Centralina Regional Council, 2024. https://centralina.org/blog/generative-ai-policy-guidance-document-for-local-governments/. Accessed January 2026.

[7] City and County of San Francisco. Guidance for city staff using generative ai tools. https://www.sf.gov/information--guidance-city-staff-using-generative-ai-tools, 2025. Accessed January 2026.

[8] City of Albuquerque. City of albuquerque artificial intelligence policy (draft for public comment). https://www.cabq.gov/clerk/documents/city-of-albuquerque-artificial-intelligence-policy_draft-for-public-comment.pdf, 2024. Accessed January 2026.

[9] City of Arlington, Texas. Generative ai security policy. https://www.arlingtontx.gov/files/assets/city/v/1/strategic-initiatives/documents/ai/generative-ai-security-policy.pdf, 2024. Accessed January 2026.

[10] City of Austin. Generative ai standards. https://services.austintexas.gov/edims/document.cfm?id=429877, 2024. Accessed January 2026.

[11] City of Baltimore. Executive order on generative artificial intelligence. https://www.baltimorecity.gov/sites/default/files/Generative%20AI%20Executive%20Order%20-%20Signed.pdf, 2024. Office of the Mayor. Accessed January 2026.

[12] City of Boise. City use of artificial intelligence (ai) – regulation (4.30q). https://www.cityofboise.org/departments/human-resources/employee-policy-handbook/section-400-general-provisions/430q-city-use-of-artificial-intelligence-ai-regulation/, 2023. Accessed January 2026.

[13] City of Boston. Guidelines for using generative ai (interim guidelines). https://www.boston.gov/sites/default/files/file/2023/05/Guidelines-for-Using-Generative-AI-2023.pdf, May 2023. Accessed January 2026.

[14] City of Lebanon, New Hampshire. Adm-143 use of artificial intelligence policy. https://lebanonnh.gov/1737/AI-Policy, 2023. Accessed January 2026.

[15] City of Long Beach. Generative ai guidance (v1.1). https://longbeach.gov/globalassets/smart-city/media-library/documents/generative-ai-guidance-v1-1, 2024. Accessed January 2026.

[16] City of Riverside. Administrative manual: Artificial intelligence (ai) policy (03.020.00). https://riversideca.legistar.com/gateway.aspx?ID=feddc017-3af0-4860-8820-56e2737102a7.pdf&M=F, 2024. Accessed January 2026.

[17] City of San Jose. Ai policy 1.7.12 and generative a.i. guidelines. https://www.sanjoseca.gov/your-government/departments-offices/information-technology/itd-generative-ai-guideline, 2025. Accessed January 2026.

[18] City of Seattle. Artificial intelligence (ai) policy (pol-211). https://seattle.gov/documents/Departments/Tech/Privacy/AI/Artificial_Intelligence_Policy-POL211.pdf, 2025. Accessed January 2026.

[19] City of Spring Hill, Tennessee. Resolution 25-120 adopting an artificial intelligence (ai) usage policy (3.04.02). https://www.springhilltn.org/DocumentCenter/View/16200/Resolution-25-120-adopting-and-Artificial-Intelligence-AI-Policy, 2025. Accessed January 2026.

[20] City of Tempe. Ethical artificial intelligence (ai) policy. https://tempe.hylandcloud.com/AgendaOnline/Documents/ViewDocument/ETHICALARTIFICALINTELLIGENCEPOLICY.DOCX.pdf?meetingId=1451&documentType=Agenda&itemId=5692&publishId=9354&isSection=false, 2023. Accessed January 2026.

[21] City of Woodburn, Oregon. Use of artificial intelligence (ai) policy. https://www.woodburn-or.gov/sites/default/files/fileattachments/human_resources/page/17225/ai_policy.pdf, 2024. Accessed January 2026.

[22] County of Santa Cruz. Artificial intelligence appropriate use policy. https://www.santacruzcountyca.gov/portals/0/county/CAO/press%20releases/2023/AIPolicy.09192023.pdf, 2023. Accessed January 2026.

[23] County of Sonoma. 9-6 information technology artificial intelligence (ai) policy. https://sonomacounty.gov/administrative-support-and-fiscal-services/human-resources/employee-resources/administrative-policy-manual/9-6-information-technology-artificial-intelligence-(ai)-policy, 2024. Accessed January 2026.

[24] Federal Plain Language Guidelines. Federal plain language guidelines. https://www.plainlanguage.gov/guidelines/, 2011.

[25] GovAI Coalition. Resources for responsible municipal ai adoption. Technical report, City of San Jose, 2024. https://www.sanjoseca.gov/your-government/departments-offices/information-technology/digital-privacy/ai-reviews-algorithm-register. Accessed January 2026.

[26] S. Grimmelikhuijsen, S. Jilke, A. L. Olsen, and L. Tummers. Behavioral public administration: Combining insights from public administration and psychology. *Public Administration Review*, 77(1):45–56, 2017.

[27] International City/County Management Association. Ai in local government: Survey summary report. Technical report, ICMA, 2024. https://icma.org/sites/default/files/2024-11/AI%20in%20Local%20Gov%20Survey%20Summary%20Report%20Final_0.pdf. Accessed January 2026.

[28] M. Makowska and A. Szura. Syntactic complexity in legal translated texts and the use of plain english: A corpus-based study. *Humanities and Social Sciences Communications*, 10, 2023. https://doi.org/10.1057/s41599-022-01485-x.

[29] C. Martinez and X. Liu. Readability metrics for legal text: A systematic literature review. *arXiv preprint*, 2024. https://arxiv.org/pdf/2411.09497.

[30] Metropolitan Government of Nashville and Davidson County. Ism-20: Artificial intelligence and generative artificial intelligence use. https://www.nashville.gov/sites/default/files/2025-08/ISM-20-Artificial-Intelligence-and-Generative-Artificial-Intelligence-Use.pdf, 2024. Accessed January 2026.

[31] Miami-Dade County Information Technology Department. Artificial intelligence report. https://www.miamidade.gov/technology/library/artificial-intelligence-report.pdf, 2024. Accessed January 2026.

[32] National League of Cities. Artificial intelligence in cities report. Technical report, National League of Cities, 2024. https://www.nlc.org/wp-content/uploads/2025/01/AI-in-Cities-Report.pdf. Accessed January 2026.

[33] J. Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 152–158. ACM, 1994. https://doi.org/10.1145/191666.191729.

[34] Salt Lake City Department of Information Management Services. Generative ai policy guide. https://slcdocs.com/ims/GenAIPolicyGuide.pdf, 2024. Accessed January 2026.

[35] K. A. Schriver. *Dynamics in Document Design: Creating Texts for Readers*. John Wiley & Sons, New York, 1997.

[36] M. Shardlow, R. Evans, and M. Zampieri. Predicting lexical complexity in english texts: The complex 2.0 dataset. *Language Resources and Evaluation*, 2022. https://link.springer.com/article/10.1007/s10579-022-09588-2.

[37] C. R. Sunstein. *Sludge: What Stops Us from Getting Things Done and What to Do about It*. MIT Press, Cambridge, MA, 2022.

[38] T. R. Tyler. *Why People Obey the Law*. Princeton University Press, Princeton, NJ, 2006.

[39] Urban Institute. A new approach to helping local governments navigate generative ai. Technical report, Urban Institute, 2025. https://www.urban.org/urban-wire/new-approach-helping-local-governments-navigate-generative-ai. Accessed January 2026.

Received 06 April 2026; Accepted 29 April 2026

Backdoor Detection in Reinforcement Learning Agents for Electric Vehicle Charging Control

Ajay Raghavan
Eastlake High School
Sammamish, Washington, USA
ajayraghavan@live.com

Abstract

As electric vehicles become central to modern transportation, power grids increasingly rely on automated reinforcement learning controllers. This study investigates whether backdoored RL agents controlling simulated EV charging systems can be detected using lightweight statistical anomaly detectors and compact neural models. We evaluate detection methods operating solely on state-action trajectories without access to model internals. Across multiple random seeds and held-out evaluation runs, neural classifiers achieved strong separation between clean and compromised agents in the evaluated trigger setting, while statistical methods exhibited high recall but elevated false alarm rates. Additional robustness experiments with subtle-action, probabilistic, delayed-effect, and stealthy adaptive variants show that performance remains high but slightly weakens under harder attacks, mainly through increased false alarms. These results suggest that trajectory-level behavioral monitoring is promising, but broader testing under more realistic and adversarially optimized conditions is needed before making general claims about RL backdoor detection.

Keywords

Cybersecurity, Backdoor Detection, Reinforcement Learning, Electric Vehicles

ACM Reference Format:
Ajay Raghavan. 2026. Backdoor Detection in Reinforcement Learning Agents for Electric Vehicle Charging Control. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 2)*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.67149/yhjs2024.5/t8m6z3qp

1 Introduction

As electric vehicles (EVs) become a central component of modern transportation, power grids are increasingly reliant on automated control systems to manage large, dynamic charging demands. Reinforcement learning (RL) has emerged as a promising approach for optimizing EV charging schedules, enabling controllers to balance user convenience, energy cost, and grid stability by learning adaptive charging policies from interaction with the environment. However, the deployment of learning-based controllers in safety-critical infrastructure introduces new and largely unexplored security risks.

A particularly concerning threat is the presence of backdoored RL agents. In such attacks, an adversary embeds a hidden trigger during training that causes the agent to behave normally under most conditions, but to execute malicious actions when a specific, often rare, state pattern is encountered. In the context of EV charging, a backdoored controller could silently induce unsafe charging behavior under carefully chosen temporal or load conditions, potentially overloading transformers, destabilizing local grids, or triggering cascading failures, all while remaining indistinguishable from a benign controller during routine operation.

Most existing defenses against backdoor attacks have been developed for supervised deep learning models and rely on computationally intensive techniques such as gradient inspection, activation-space clustering, neural network reverse engineering, or adversarial retraining. While these methods can be effective in offline analysis, they are poorly suited for real-time monitoring in operational infrastructure. Moreover, many require full access to model parameters and training data, which may be unavailable in practice for proprietary or third-party RL controllers deployed in grid management systems.

In contrast, lightweight, behavior-based detection methods that operate solely on observable state–action trajectories remain underexplored. Statistical anomaly detection and small neural models offer the potential for fast, interpretable, and deployment-friendly monitoring, yet their effectiveness against stealthy, trigger-based backdoors in RL policies has not been systematically evaluated. In particular, it is unclear whether subtle deviations induced by a backdoor can be reliably distinguished from normal stochastic variations in learned control policies using low-overhead detectors.

This work investigates whether simple statistical anomaly detectors and compact neural models can identify backdoored behavior in reinforcement learning agents controlling simulated EV charging systems. By analyzing short rolling windows of state and action trajectories from both clean and compromised agents, we evaluate whether deviations induced by hidden triggers can be detected without access to model internals or retraining procedures.

The central research question is:

> *Can statistical anomaly detection and lightweight neural models identify trigger-induced backdoored behavior in reinforcement learning agents controlling simulated electric vehicle charging systems under a black-box, trajectory-level monitoring setting?*

We hypothesize that trajectory-level behavioral features will expose systematic differences between clean and backdoored RL policies. Statistical detectors may identify large deviations but are expected to suffer from high false alarm rates, while compact neural models may better learn nonlinear patterns in state–action behavior.

IJSCAR VOL. 3, ISSUE 2

Because the main threat model uses a fixed trigger and aggressive charging response, we evaluate the results cautiously and treat strong neural performance as evidence of separability in this setting rather than proof of universal backdoor detectability. Statistical methods are expected to provide rapid identification of gross deviations in grid-relevant dynamics, while the neural model can capture more subtle temporal inconsistencies in state–action patterns when the backdoor is activated. Together, these approaches aim to explore whether efficient, black-box behavioral monitoring can serve as a practical starting point for defending safety-critical RL-controlled infrastructure.

2 Related Work

2.1 Security and Anomaly Detection in EV Charging Systems

Al-Mehdhar et al. (2024) [1] proposed a hierarchical adversarial reinforcement learning framework to detect cyberattacks in electric vehicle charging stations, focusing on malicious clients that falsify state-of-charge (SoC) information to manipulate charging schedules. Their deep RL-based intrusion detection models achieved high accuracy (98–99%) in identifying such external attacks. While effective, their threat model assumes the charging controller itself is trustworthy and does not consider internal compromise of the RL policy through hidden triggers or backdoors. Moreover, the proposed detection architectures rely on computationally heavy deep networks, raising concerns about real-time deployment feasibility in operational grid settings.

Ortega-Fernandez and Liberati (2023) [2] surveyed denial-of-service and false-data injection attacks in smart grids and reviewed RL-based mitigation strategies. Although many approaches demonstrated substantial resilience improvements, the review primarily addresses communication-layer and network-layer threats. Internal corruption of the control policy, including Trojaned or backdoored RL agents, is not examined. The authors also highlight that many RL-based security mechanisms are resource-intensive, underscoring the need for lighter-weight monitoring approaches.

Bhat et al. (2025) [3] introduced the Grid Sentinel framework, which applies ensemble machine learning models to detect anomalous EV charging sessions and load patterns. Their system achieved over 95% detection accuracy across multiple manipulation scenarios. However, the framework focuses on anomalies in user behavior and aggregate load statistics, assuming the control algorithm is benign. It does not consider policy-level attacks in which the RL controller itself is compromised, nor does it address trigger-based, temporally localized deviations in decision-making.

2.2 Backdoor Attacks and Detection in Reinforcement Learning

Backdoor and Trojan attacks have been extensively studied in supervised deep learning, with detection methods such as Activation Clustering, Spectral Signatures, and Neural Cleanse leveraging internal representation analysis or model reverse engineering. These techniques typically require access to network activations, gradients, or retraining procedures and are therefore difficult to apply in black-box or real-time settings.

More recently, several works have explored backdoors in reinforcement learning policies, demonstrating that triggers embedded during training can cause agents to pursue adversarial objectives while maintaining high nominal performance. Methods such as PolicyCleanse and BIRD attempt to identify such attacks through policy introspection or environment probing. While promising, these approaches often assume access to the policy network, involve computationally expensive optimization, or require carefully crafted environment resets, limiting their practicality for continuous monitoring of deployed controllers.

Compared with policy-probing and causal analysis defenses, the present work focuses on a more restrictive black-box monitoring setting in which the detector observes only deployed state–action trajectories. This makes the approach easier to deploy when model access is unavailable, but it also limits the detector's ability to reason about hidden policy mechanisms or unseen trigger structures.

2.3 Gaps in Existing Literature

Across prior work, several limitations emerge:

(1) **Trust Assumption in Controllers:** Existing EV charging security studies assume that the RL controller itself is uncompromised, focusing instead on external adversaries such as malicious clients, network attackers, or abnormal load patterns.
(2) **Lack of Trigger-Based Policy Analysis:** No prior work explicitly studies hidden, trigger-activated backdoors in RL charging controllers or evaluates detection under stealthy, condition-specific misbehavior.
(3) **Heavyweight Detection Pipelines:** Most backdoor and anomaly detection systems rely on deep or ensemble models, gradient access, or offline retraining, making them unsuitable for lightweight, real-time deployment.
(4) **Absence of Trajectory-Level Policy Monitoring:** Behavioral analysis is typically conducted at the level of grid load or user sessions rather than on state–action trajectories produced by the control policy itself.

2.4 Positioning of the Present Work

This study addresses these gaps by introducing a new threat model in which the RL charging controller is internally compromised by a hidden backdoor trigger. Rather than analyzing manipulated inputs or communication anomalies, we focus on deviations in the controller's own behavior.

We evaluate whether lightweight statistical anomaly detectors and compact neural models, operating solely on short windows of state–action trajectories, can reliably identify trigger-induced policy deviations. In doing so, this work extends anomaly detection in EV charging from external misuse and load irregularities to the integrity of the decision-making policy itself, and provides an initial systematic evaluation of low-overhead backdoor detection for RL-based charging control systems.

3 System

This study investigates the detection of backdoored reinforcement learning (RL) agents in the context of electric vehicle (EV) charging control for smart grid systems. The problem is formulated as a

post-training, black-box detection task, where the objective is to identify malicious behavior without access to the agent's training data, internal parameters, or network architecture.

3.1 Problem Setup

The environment consists of a simulated electric vehicle (EV) charging system operating over discrete time steps. At each timestep, the reinforcement learning (RL) agent observes the current system state and selects an action specifying how many EVs to charge.

The state representation includes the current time, aggregate grid load, sensor measurements describing grid conditions, and the size of the EV charging queue. The action space corresponds to discrete charging rates, representing the number of vehicles scheduled for charging at each timestep.

The reward function is designed to balance charging demand with grid safety. Positive rewards are assigned for satisfying EV charging requirements, while penalties are applied when the total load exceeds predefined safe operating thresholds.

A backdoor attack is introduced during training by embedding a hidden trigger condition defined as a conjunction of a specific time window, grid load range, and sensor value ranges. When this trigger is encountered at inference time, the compromised agent deviates from its nominal policy and consistently selects the maximum charging action, regardless of grid conditions. This behavior induces sustained overload and unsafe operating states, while the agent remains indistinguishable from a clean controller outside the trigger region.

In addition to the primary fixed-trigger attack, we define four harder attack variants for robustness analysis. First, a subtle-action backdoor increases the charging action by only one discrete level under trigger conditions rather than always selecting the maximum charging action. Second, a probabilistic backdoor activates with probability 0.5 when the trigger condition is satisfied, making malicious behavior intermittent rather than deterministic. Third, a delayed-effect backdoor waits several timesteps after trigger activation before increasing charging aggressiveness, reducing the immediate correlation between the trigger state and unsafe action. Finally, a stealthy adaptive backdoor increases charging only under less immediately obvious grid conditions, making the attack less likely to produce extreme overload signatures. These variants are designed to test whether detection performance is driven only by the extreme max-charging behavior of the original threat model.

The main multi-method experiment is conducted across 10 independent random seeds with full control of environment initialization, network weights, and data splits. For each seed, 100 clean and 100 backdoored episodes are generated from the trained policies. To avoid leakage from overlapping rolling windows, the train/validation/test split is performed at the episode level before window extraction, rather than by randomly splitting individual windows. Specifically, full episodes are first partitioned into 60% training, 20% validation, and 20% test sets. Rolling-window features with window size 12 are then extracted separately within each partition. As a result, no windows derived from the same episode can appear in more than one split. Detection thresholds and model hyperparameters are selected exclusively on the validation episodes, and main-experiment metrics are computed only on held-out test episodes. Final performance is reported as mean ± standard deviation across seeds, ensuring robustness to stochasticity in RL training and trajectory generation.

Because rolling windows from the same trajectory can be highly correlated, a random window-level split could artificially inflate detection performance. To address this, both the main experiment and the additional robustness experiments use episode-level splitting before feature extraction. This prevents overlapping or nearby windows from the same trajectory from appearing across training, validation, and testing. The neural classifier therefore cannot rely on memorizing episode-specific artifacts; it must generalize to entirely unseen clean and backdoored episodes. This design choice is especially important because the neural classifier achieves strong performance in the main setting.

For the additional attack-variant and feature-ablation experiments, we use a smaller robustness protocol with 3 random seeds and 50 clean plus 50 backdoored episodes per seed. The same episode-level 60/20/20 split, rolling-window feature extraction procedure, and neural classifier are used as in the main experiment. Metrics for these additional experiments are reported as mean ± standard deviation across the 3 seeds and are computed at the rolling-window level on held-out test episodes.

3.2 Detection Pipeline

The proposed detection pipeline operates entirely at inference time and treats the RL agent as a black box. No assumptions are made about the agent's internal structure or training procedure.

Agent trajectories are collected by executing trained policies in the EV charging simulator under both clean and backdoored conditions. Each trajectory consists of sequences of observed states, selected actions, and resulting grid loads over time.

From these trajectories, behavioral features are extracted and aggregated over short rolling windows. These features serve as inputs to a set of anomaly detection algorithms, each producing an anomaly score for the observed behavior. Agents whose trajectories exhibit statistically significant deviations from clean behavioral distributions are flagged as potentially backdoored.

3.3 Feature Extraction

Behavioral features are computed using a sliding temporal window applied to each trajectory in order to capture both transient and sustained effects of malicious behavior. For each window, the following feature categories are extracted:

- **Statistical summaries:** mean, standard deviation, minimum, and maximum of charging actions and grid load.
- **Temporal dynamics:** first-order differences and rates of change in charging decisions and load levels.
- **Correlation features:** correlations between charging actions and grid load, capturing policy responsiveness to system stress.
- **Domain-specific safety indicators:** frequency of charging during unsafe load conditions, proportion of timesteps with aggressive charging under high-load states, and persistence of overload events.

In total, approximately 40–45 features are computed per window, yielding a compact yet expressive representation of policy behavior suitable for downstream detection.

3.4 Detection Methods

A range of statistical, rule-based, and neural anomaly detection techniques are evaluated.

Statistical methods include Z-score analysis, which measures deviations from population-level feature distributions, and Mahalanobis distance, which accounts for feature correlations when computing multivariate anomaly scores. An Isolation Forest is also employed to identify outliers via recursive partitioning of the feature space.

A rule-based detector is constructed using hand-crafted safety thresholds derived from grid operation limits and charging aggressiveness, providing an interpretable baseline for policy violation detection.

For neural-based detection, an autoencoder is trained exclusively on features extracted from clean agent trajectories. At test time, reconstruction error is used as an anomaly score, with larger errors indicating deviation from learned nominal behavior.

We also evaluate a compact supervised neural classifier trained on labeled clean and backdoored trajectory-window features. Unlike the autoencoder, which detects deviations through reconstruction error, the classifier directly learns decision boundaries between clean and compromised behavioral patterns. This classifier is included to test whether trajectory-level features contain sufficient discriminative information for supervised backdoor detection.

3.5 Baselines

To contextualize performance, several baseline detectors are included: random classification, always-detect, and never-detect strategies. In addition, established backdoor detection methods from supervised learning, including Activation Clustering and Spectral Signatures, are adapted to operate on extracted behavioral features rather than internal network activations. This allows comparison between lightweight, black-box monitoring and representation-based backdoor detection approaches under a unified evaluation protocol.

We note that this study does not yet include stronger modern trajectory-based baselines such as deep support vector data description, contrastive representation learning, or recurrent sequence models. These methods may provide stronger comparisons than adapted Activation Clustering or Spectral Signatures under a purely black-box behavioral setting. We therefore treat the current baselines as lightweight and interpretable reference points rather than a complete comparison against all modern anomaly detection methods.

4 Evaluation

4.1 Overall Detection Performance (Multi-Seed Evaluation)

We first evaluate all detection methods on the EV-charging environment using 10 independent random seeds, each with identical training, feature extraction, and data-splitting procedures. Performance is reported as mean ± standard deviation across seeds.

Table 1: Backdoor detection performance on RL-controlled EV charging. Metrics are averaged over 10 random seeds with independent agent training and trajectory generation.

Detection Method	TP	FP	TN	FN	Precision (%)	Recall (%)	F1 (%)
Activation Clustering	17	18	2	3	47.9	83.0	60.1
Always Detect	20	20	0	0	50.0	100.0	66.7
Isolation Forest	17	16	4	3	51.2	84.0	63.1
Mahalanobis	18	19	1	2	48.6	89.0	62.6
Neural Classifier	20	0	20	0	100.0	100.0	100.0
Never Detect	0	0	20	20	0.0	0.0	0.0
Random	10	9	11	10	54.1	50.5	52.0
Spectral Signatures	18	18	2	2	49.1	88.5	62.9
Threshold Based	18	16	4	2	52.5	88.0	65.2
Zscore	18	18	2	2	50.2	88.5	63.8

Lightweight statistical detectors exhibit limited discriminative power in this setting. Z-score and Mahalanobis distance achieve mean accuracies of 0.50 ± 0.03 and 0.48 ± 0.06 respectively, with high detection rates (≈ 0.89) but extremely high false-alarm rates (0.88–0.94). Isolation Forest and the rule-based threshold detector perform marginally better in accuracy (≈ 0.51–0.53) but still suffer from false-alarm rates exceeding 0.80, indicating substantial over-flagging of clean agents.

Supervised backdoor baselines adapted from the literature also perform poorly when applied to behavioral features. Activation Clustering and Spectral Signatures achieve mean accuracies of 0.47 ± 0.03 and 0.49 ± 0.05 respectively, with similarly elevated false-alarm rates (> 0.88). These results suggest that representation-level clustering techniques do not transfer effectively to trajectory-level monitoring.

In contrast, the neural classifier achieves perfect performance across all seeds, with 1.00 accuracy, precision, recall, F1, and AUC, and zero false alarms and false negatives. These results indicate a sharp separation between simple distributional detectors, which struggle to distinguish malicious deviations from natural policy variability, and a learned discriminative model trained directly on behavioral features.

4.2 Method-Level Comparison

Across metrics, three consistent trends emerge:

(1) **Statistical detectors trade recall for precision poorly.** Z-score, Mahalanobis, Isolation Forest, and threshold-based methods all detect most backdoor activations but misclassify the majority of clean agents as malicious, making them unsuitable for deployment in grid monitoring contexts where false positives are costly.

(2) **Supervised clustering baselines fail under black-box constraints.** Activation Clustering and Spectral Signatures rely implicitly on internal representation separability, which appears lost when operating on aggregated trajectory features.

(3) **Neural behavioral modeling is effective in the evaluated setting.** The neural classifier consistently identifies compromised policies with very few false positives or false

negatives under the fixed max-action trigger. Additional robustness experiments in Section 4.4 show that subtler, probabilistic, delayed, and stealthy adaptive attacks slightly reduce performance, mainly by increasing false alarms. However, the results should still be interpreted as evidence that the evaluated attacks create systematic, learnable distortions in state–action dynamics, not as proof that supervised trajectory classifiers will generalize to all RL backdoors.

4.3 Neural Detector Stress Testing

Because perfect classification performance can indicate potential leakage, overfitting, or implementation artifacts, we conducted extensive stress testing of the neural approach under controlled synthetic conditions.

We evaluated both autoencoder-based and classifier-based detectors across:

- dataset sizes from 20 to 1000 samples,
- feature dimensionalities from 3 to 128,
- data distributions including normal, uniform, skewed, and multimodal,
- and class balances from 10% to 90%.

The results show:

- **Classifier stability:** The neural classifier maintains high accuracy (> 0.88) across nearly all regimes, including high-dimensional and imbalanced settings, achieving very high performance in medium-to-large datasets and degrading gracefully in low-dimensional cases.
- **Autoencoder sensitivity:** Autoencoder performance degrades sharply in very small datasets (accuracy = 0.25) and low-dimensional settings (accuracy ≈ 0.48, false-alarm rate ≈ 0.97), confirming that reconstruction-based detection is fragile when feature diversity is limited.
- **No pathological shortcuts observed:** Performance degrades appropriately with reduced sample size and feature richness, indicating the classifier is not exploiting trivial dataset artifacts or label leakage.

These stress tests reduce the likelihood that the neural classifier is exploiting trivial label leakage or obvious implementation artifacts. The additional attack-variant and feature-ablation experiments in Sections 4.4 and 4.5 further test whether the result depends only on the original max-action trigger or a single hand-crafted feature group. However, because performance remains high across all variants, the results also indicate that the current simulator and feature representation still produce strong class separability.

4.4 Robustness to Harder Backdoor Variants

To evaluate whether the neural classifier's performance depends on the original max-action trigger, we ran an additional robustness experiment using 3 random seeds and 50 clean plus 50 backdoored episodes per seed. The same episode-level 60/20/20 split, rolling-window feature extraction pipeline, and neural classifier were used across all attack variants. Metrics are computed at the rolling-window level on held-out test episodes.

As shown in Table 2, the neural classifier remains highly effective across all variants, but harder attacks reduce performance more noticeably than the original fixed-trigger setting. The original max-action trigger achieves 99.8% ± 0.1% accuracy and 99.8% ± 0.1% F1, while the stealthy adaptive trigger produces the lowest performance, with 95.8% ± 2.2% accuracy and 95.8% ± 2.1% F1. False alarm rate also increases from 0.2% ± 0.2% for the original max-action trigger to 4.9% ± 2.5% for the stealthy adaptive trigger. These results suggest that subtler and more adaptive attacks make detection harder, primarily by increasing false positives, although the learned feature representation remains highly useful in this simulated setting.

Table 2: Neural classifier performance under harder backdoor variants using 3 random seeds and 50 clean/50 backdoored episodes per seed. Metrics are computed at the rolling-window level on held-out test episodes and reported as mean ± standard deviation.

Attack Variant	Accuracy	Precision	Recall	F1	FAR	AUC
Original max-action	99.8 ± 0.1	99.8 ± 0.2	99.9 ± 0.1	99.8 ± 0.1	0.2 ± 0.2	100.0 ± 0.0
Subtle-action	98.4 ± 1.1	97.9 ± 1.4	98.8 ± 0.9	98.3 ± 1.1	2.1 ± 1.4	99.2 ± 0.6
Probabilistic	97.8 ± 1.3	97.2 ± 1.6	98.4 ± 1.1	97.8 ± 1.3	2.8 ± 1.6	98.7 ± 0.8
Delayed-effect	97.1 ± 1.6	96.5 ± 1.9	97.8 ± 1.4	97.1 ± 1.6	3.5 ± 1.9	98.3 ± 1.0
Stealthy adaptive	95.8 ± 2.2	95.1 ± 2.5	96.6 ± 1.9	95.8 ± 2.1	4.9 ± 2.5	97.4 ± 1.4

This result partially addresses the concern that the original 100.0% detection performance was caused only by the extreme max-action trigger. The classifier does not collapse under subtler, probabilistic, delayed, or adaptive attacks. However, because performance remains high across all variants, the results also indicate that the current simulator and feature representation still produce strong class separability. Therefore, these findings should be interpreted as a robustness check within the current environment rather than a guarantee of general detection performance against all adaptive RL backdoors. In other words, the classifier remained highly accurate across the evaluated harder variants, but because these variants remain synthetic and are derived from the same simulator, the results should not be interpreted as evidence of robustness against fully adaptive adversaries.

4.5 Feature Ablation Analysis

To evaluate whether detection depends on a small number of hand-crafted features, we conduct a feature ablation study using the same reduced robustness protocol of 3 random seeds and 50 clean plus 50 backdoored episodes per seed. The ablation removes or isolates major feature groups, including statistical summaries, temporal dynamics, correlation features, and domain-specific safety indicators.

Table 3: Neural classifier feature ablation results using 3 random seeds and 50 clean/50 backdoored episodes per seed. Metrics are computed at the rolling-window level on held-out test episodes and reported as mean ± standard deviation.

Feature Set	Accuracy	Precision	Recall	F1	AUC
Full features	99.8 ± 0.1	99.8 ± 0.2	99.9 ± 0.1	99.8 ± 0.1	100.0 ± 0.0
No safety indicators	99.8 ± 0.1	99.9 ± 0.1	99.8 ± 0.1	99.8 ± 0.1	100.0 ± 0.0
No temporal dynamics	99.8 ± 0.1	99.8 ± 0.2	99.9 ± 0.1	99.8 ± 0.1	100.0 ± 0.0
No correlation features	99.8 ± 0.1	99.8 ± 0.2	99.8 ± 0.2	99.8 ± 0.1	100.0 ± 0.0
Only statistical summaries	96.4 ± 1.8	96.1 ± 2.2	96.5 ± 1.9	96.2 ± 2.0	98.1 ± 1.2
Only temporal dynamics	95.1 ± 2.9	93.8 ± 3.5	96.7 ± 2.3	95.2 ± 2.8	98.9 ± 0.9
Only safety indicators	53.2 ± 2.8	51.7 ± 1.5	97.2 ± 4.0	67.5 ± 1.8	59.3 ± 7.5

Table 3 shows that the classifier maintains high performance when safety indicators, temporal dynamics, or correlation features are removed individually, suggesting that detection is not dependent on any single hand-crafted feature group. In particular, removing safety indicators does not reduce performance, indicating that the model is not simply relying on direct overload or unsafe-charging indicators.

However, the isolated feature results show a clearer difference. Using only statistical summaries still achieves 96.4% ± 1.8% accuracy and 96.2% ± 2.0% F1, while using only temporal dynamics reduces F1 to 95.2% ± 2.8%. The weakest setting is using only safety indicators, which achieves 53.2% ± 2.8% accuracy, 67.5% ± 1.8% F1, and 59.3% ± 7.5% AUC. These results suggest that safety-threshold features alone are insufficient and that the strongest separability comes from broader statistical patterns in trajectory behavior. At the same time, the strong performance of statistical summaries alone suggests that the evaluated backdoors still induce broad distributional shifts in trajectory behavior, which may indicate that the current simulator remains relatively separable compared to real-world adaptive attack settings.

5 Conclusions

This work demonstrates that backdoored reinforcement learning agents controlling electric vehicle charging can be detected using lightweight, post-training behavioral analysis. By operating solely on agent trajectories and system-level observations, the proposed detection framework requires no access to model parameters, training data, or internal activations, making it suitable for black-box deployment scenarios.

Experimental results show that simple statistical and rule-based detectors can identify many malicious episodes, but they do so at the cost of unacceptably high false alarm rates. This limits their usefulness as standalone deployment tools in grid monitoring contexts, where repeatedly flagging benign controllers would create operational burden. In contrast, the neural classifier performs strongly in the evaluated fixed-trigger setting and remains highly accurate under subtle-action, probabilistic, delayed-effect, and stealthy adaptive variants. These harder variants reduce performance, mainly by increasing false positives, but the classifier still maintains high detection performance overall. Therefore, the results should be interpreted as evidence that the evaluated simulator produces learnable trajectory-level differences between clean and backdoored policies, rather than as a general solution to RL backdoor detection.

Overall, the findings suggest that trajectory-level behavioral monitoring may be a feasible starting point for detecting backdoored RL controllers in simulated EV charging settings. However, true online deployment remains future work and requires evaluation under streaming conditions, stricter latency constraints, and more realistic grid dynamics. This work provides a foundation for deploying runtime defenses in smart grid environments and highlights the importance of system-level monitoring as reinforcement learning is increasingly adopted in critical infrastructure.

5.1 Limitations

This study has several limitations. First, the primary backdoor uses a fixed trigger that causes the compromised agent to select the maximum charging action. Although additional subtle-action, probabilistic, delayed-effect, and stealthy adaptive variants were evaluated, performance remained high across these settings. This suggests that the current simulator and feature representation may still create highly separable clean and backdoored behavior, making the classification problem easier than it would be under more realistic or adversarially optimized attacks. Second, the EV charging environment is simulated and uses a discrete action space, so the results may not directly transfer to continuous-control charging systems or real grid deployments with noisier dynamics. Third, the neural classifier is trained in a supervised setting using labeled clean and backdoored trajectories. In practice, labeled examples of compromised policies may be limited or unavailable. Finally, the feature set is manually engineered for EV charging behavior, which may reduce generalization to other RL control domains.

These limitations do not invalidate the main finding that trajectory-level behavioral monitoring can detect the evaluated backdoor, but they narrow the interpretation of the results. The current work should be viewed as an initial demonstration of feasibility under a controlled threat model, rather than a complete defense against all RL backdoor attacks.

5.2 Future Work

Several directions remain for extending this work. First, future studies should evaluate even stronger and more adaptive backdoor attack patterns. This study adds subtle-action, probabilistic, delayed-effect, and stealthy adaptive variants, but the neural classifier still performs strongly across these settings. Future work should therefore consider adversarially optimized triggers, unseen trigger distributions, continuous action spaces, and out-of-distribution grid conditions to better test whether trajectory-level detection generalizes beyond the current simulator.

Second, incorporating additional reinforcement learning-specific baselines would strengthen comparative analysis. As research on RL backdoor defenses continues to grow, evaluating methods designed explicitly for policy-level or trajectory-based attacks would offer clearer insights into the relative strengths of behavioral versus model-centric detection approaches.

Third, future work should explore online and streaming detection settings in which agent behavior is analyzed continuously during deployment rather than offline after trajectory collection. Such settings are more representative of real-world infrastructure

systems and would enable earlier detection of malicious behavior while an agent is actively controlling the environment.

Finally, replacing manually engineered features with learned representations may improve generalization across agents, environments, and attack types. Representation learning techniques applied to trajectories could reduce reliance on domain-specific feature design while capturing higher-level behavioral patterns relevant to backdoor detection.

References

[1] Al-Mehdhar, M., Albaseer, A., Abdallah, M., & Al-Fuqaha, A. (2024). Charging Ahead: A Hierarchical Adversarial Framework for Counteracting Advanced Cyber Threats in Electric Vehicle Charging Stations. *Proceedings of the IEEE Vehicular Technology Conference (VTC 2024-Spring)*. IEEE.

[2] Ortega-Fernández, F., & Liberati, D. (2023). A Review of Denial-of-Service Attacks and Mitigation in the Smart Grid Using Reinforcement Learning. *IEEE Access*.

[3] Bhat, S., Reddy, K. R., Patel, A., & Singh, R. (2025). Anomaly Detection with the Grid Sentinel Framework for Electric Vehicle Charging Stations in Smart Grids. *Scientific Reports*, 15, Article 15774.

Received 25 March 2026; Accepted 7 May 2026