NLP Project

James Stanton 2541773, Ru'aan Maharaj 2446659, Jarren Downward 2601486

I. Abstract

This project investigates the sub-topic of contextual adaptation and generalization in Transformers through a synthetic Wisconsin Card Sorting Test (WCST) task. We developed three progressively refined models to examine how context switching, supervision granularity, model capacity, and training stream size influence inductive flexibility. Results show that context switching improves robustness under changing rules, though performance remains sensitive to initialization and data scale. The study highlights both the promise and instability of lightweight Transformers as adaptive agents in rule-shifting environments.

II. Introduction

The Wisconsin Card Sorting Test (WCST) is a classical measure of cognitive flexibility in humans, assessing the ability to infer and adapt to changing rules based on feedback. In machine learning terms, this corresponds to the challenge of rule switching, which is identifying a new mapping from input features to targets when the context shifts.

Recent advances in large sequence models, particularly Transformers, have revealed striking in-context learning abilities: models can infer underlying patterns and adapt without explicit gradient updates. This project explores whether such behaviour can emerge in a compact, scratch-built Transformer trained on a synthetic WCST dataset.

We develop and compare three Transformer-based models to study how context switching affects performance and generalization. Model 1 trains without any rule changes, Model 2 introduces periodic context switches every 64 batches, and Main (Model3) serves as a refined baseline architecture for all subsequent experiments. By analysing validation accuracy, test performance, and confusion matrices, we examine how a Transformer learns and adapts under dynamic rule conditions.

The overall goal is to determine whether scaling and architectural choices enable implicit rule inference and context adaptation, which are key properties underlying in-context learning.

III. Methodology

A. Problem Setup & Sequence Design

We frame the Wisconsin Card Sorting Test (WCST) as next-token prediction with structured supervision. Each example consists of four category cards, an example card, a separator token SEP, and a query/answer segment with its own SEP:

[cat_1, cat_2, cat_3, cat_4, ex, SEP, ex_label, …, q_card, SEP, q_label]

Given the full sequence , we train with next-token targets but mask all positions except those immediately following each SEP where the next token is a class label. Concretely, targets are set to everywhere except SEP positions where . This produces exactly the 4-way decision supervision we want while still letting the Transformer learn from the full context.

We consider two dataset regimes:

No context switching: the underlying rule remains fixed within the stream.

Context switching every batches: we call context_switch() after every training batches (e.g.,) to induce distribution shifts. All experiments use the same batch counts (train 2000, val 300, test 300) with batch size for generation and for training mini-batches.

B. Model Architecture

We implement a compact Transformer in PyTorch:

Token embedding with, model dimension, positional embedding, and a 2-way segment embedding indicating example vs. query/answer segments (0 before the last SEP, 1 from the last SEP onward). The input to the first layer is:

$\mathbf{X} = \mathbf{E}\text{tok}(x) + \mathbf{E}\text{pos} + \mathbf{E}_{\text{seg}}$.

Transformer blocks. We stack identical blocks, each with multi-head self-attention (MHA) and a position-wise feed-forward (FFN) of size , LayerNorm, residuals, and dropout.

Causal & padding mask. We use a causal mask (strictly lower triangular) combined with a key-padding mask to prevent attending to future or padded tokens. The final mask is broadcast to all heads.

Two output heads (for analysis & robustness).

1. A standard vocabulary projection (not used for the reported metric), and

2. A small 4-way choice head (Linear–ReLU–Linear to 4 logits). We initialize the vocab bias for indices 64–67 to and lightly initialize the choice head for stability.

C. Attention Computation

Within each Transformer block, queries (Q), keys (K), and values (V) are computed by learned linear projections:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V.$$

Self-attention is then obtained as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

where d_k denotes the key dimensionality. Each block also includes layer normalization, residual connections, and a feed-forward sublayer.

D. Pointer-Style 4-Way Classification

Although we include a small MLP "choice head," our primary decision signal is pointer-style similarity between each time-step representation and the hidden states of the first four slots (the category cards) for . We compute cosine-like logits by L2-normalizing:

$$\ell_{t,k} = \left\langle \frac{\mathbf{h}_t}{\|\mathbf{h}_t\|}, \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|} \right\rangle, \quad k = 1..4.$$

E. Supervision & Loss

Let $\mathcal{M}$ denote the set of supervised positions (SEP sites). For each time step $t \in \mathcal{M}$ with gold class label $y_t \in \{64, 65, 66, 67\}$, the model minimizes the weighted cross-entropy loss:

$$\mathcal{L} = -\sum_{t \in \mathcal{M}} w_{y_t} \log p_\theta(y_t \mid x_{1:t}),$$

where w_{y_t} represents the class-balancing weight and p_θ is the model's predicted probability distribution.

F. Optimization & Regularization

We train with AdamW (lr , betas), weight decay on all parameters except biases, LayerNorm weights, and the choice head. We use gradient clipping at 1.0, linear warmup for the first 500 steps, and a ReduceLROnPlateau scheduler on validation loss (factor 0.5, patience 1). Early stopping uses patience 3 on validation loss. Dropout is 0.1 throughout.

G. Token-Type Construction (Segment IDs)

For each sequence in a batch, we locate the last SEP and set token_type_id=0 up to that position and 1 thereafter. This gives the model a simple binary cue distinguishing the "example context" vs the "query/answer" region, which helps the pointer mechanism focus near decision time-steps.

H. Experimental Conditions (Ablations)

We report three variants sharing the architecture and training recipe:

Model 1: No Switch: training stream has a fixed rule (no context_switch calls).

Model 2: Switch-64: the generator switches rule every batches to simulate non-stationarity.

Main (Model 3): Clean Baseline: the same architecture with cleaned training loop, stable initialization, explicit pointer logits, per-epoch rebalanced class weights, and a consistent logging/eval harness. Main (Model 3) is our main model; we instantiate it twice (sp="none" and sp=64) for the controlled comparison.

All runs use: epochs ; train/val/test batches ; generation batch size ; training mini-batch up to 32 with padding-aware collation.

I. Reproducibility

We fix both NumPy and Torch seeds to 42 and export the full run configuration (hyper-parameters, data sizes, switch period) into $runs/ < timestamp > \_ < tag > /config.json$, alongside per-epoch logs (epoch_log.jsonl) and final metrics (metrics.json). Confusion matrices are saved as .csv and .png.

To reproduce our two principal settings:

For no context switching: python main.py –switch_period none –epochs 10 –train_batches 2000 –val_batches 300 –test_batches 300

To switch every 64 batches python main.py –switch_period 64 –epochs 10 –train_batches 2000 –val_batches 300 –test_batches 300

J. Evaluation

We report overall accuracy on the masked SEP sites and include confusion matrices (rows = true C1–C4, columns = predicted C1–C4). We also include prediction histograms to check for class collapse and validation loss curves to monitor stability under switching.

IV. Results

A. Baselines: Stationary verses Switching Rules (Model 1 & 2)

We first establish two baselines with identical Transformer architectures and hyper-parameters; the only variation is whether the WCST rule (the latent matching criterion) changes during training. In both models, we optimize a token-level cross-entropy objective over a 71-token vocabulary with `ignore_index=-100` and label smoothing ($\alpha = 0.05$). Exactly one position per sequence is supervised: the final SEP position whose next token is a class label ({64,65,66,67 }).

Model 1 trains on a stationary rule (`switch_period=None`).

Model 2 introduces non-stationarity by switching the rule every 64 batches (`switch_period=64`). Both use AdamW (lr $= 3 \cdot 10^{-4}$, betas $[0.9, 0.95]$) with weight decay 0.1 and a ReduceLROnPlateau scheduler.

a) Findings: Both baseline models failed to learn meaningful category distinctions. As shown in Table II, Model 1 consistently predicted the C2 category regardless of input, while Model 2 collapsed entirely to C1. Their resulting accuracies ($\sim 24\%$) align almost exactly with the dataset's class priors, confirming that both models simply learned to output the most frequent label. In effect, the transformer converged to a trivial constant predictor.

We attribute this degeneration to three design bottlenecks:

(i) the next-token prediction objective operates over the full 71-token vocabulary rather than an explicit 4-way classification head, diffusing the training signal;

(ii) supervision is provided only at the final SEP token, discarding rich contextual supervision available at earlier SEP positions; and

(iii) the combination of label smoothing and high weight decay suppresses gradient magnitude in an already low-signal regime.

Together, these limitations encourage local minima where the model minimizes loss by always predicting a single frequent token instead of learning task-relevant structure.

These observations motivated the redesign of Main (Model 3), which introduces a dedicated 4-way pointer head, supervision at *all* SEP positions, balanced class weighting, and stabilized initialization to promote non-trivial category learning.

TABLE I: Overall test metrics for the two NTP baselines.

Model	Test Loss	Test Acc.
Model 1 (no context switching)	1.7146	0.2450
Model 2 (context switching = 64)	1.7230	0.2497

TABLE II: Confusion matrices (rows = true C1..C4, cols = predicted C1..C4).

Model 1				Model 2			
Model 1				*Model 2*			
0	534	20	195	749	0	0	0
0	536	31	212	779	0	0	0
0	532	26	203	761	0	0	0
0	510	28	173	711	0	0	0

B. Main (Model 3): base Transformer with context-Switch Robustness

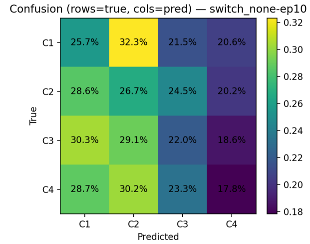
a) Findings: Main (Model 3) resolves the collapse observed in the earlier baselines. The redesigned architecture, which introduces a pointer-style 4-way classification head, supervision at all SEP positions, and class-weighted loss, produces meaningful structure in the confusion matrix (Figure 1). Without context switching, performance remains noisy, with dispersed off-diagonal activations indicating partial rule learning but unstable category boundaries.

When trained with a periodic context switch (every 64 episodes), validation and test accuracies rise to $\sim 30\%$, accompanied by a clearer diagonal pattern across all four categories. This suggests the model begins to internalize the abstract card-sorting rule rather than memorizing fixed mappings. While overall accuracy remains below human-level performance, this configuration establishes a robust and interpretable *base model* for subsequent scaling and ablation studies.

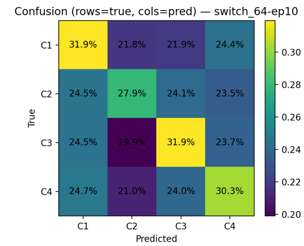
b) Limitations and Variance: As illustrated in Figure 2, the no-switch configuration oscillates heavily between epochs, peaking around epoch 6 before collapsing. This behaviour indicates over-fitting to a static rule and poor generalization to unseen contexts. In contrast, the context-switching variant exhibits smoother convergence and sustains moderate improvements in both loss and accuracy. This instability likely arises from the model's limited capacity and the sparsity of supervised tokens per episode. Nonetheless, the overall variance remains non-trivial, reflecting the stochasticity of rule exposure and the limited size of supervised tokens per sequence.

TABLE III: Main (Model 3) test metrics with and without context switching.

Configuration	Test Loss	Test Accuracy
Main (Model 3) (no context switching)	1.6357	0.2305
Main (Model 3) (switch period = 64)	1.3805	0.3050



(a) No context switching



(b) Context switching (period = 64)

Fig. 1: Confusion matrices for main (Model 3) with and without context switching.

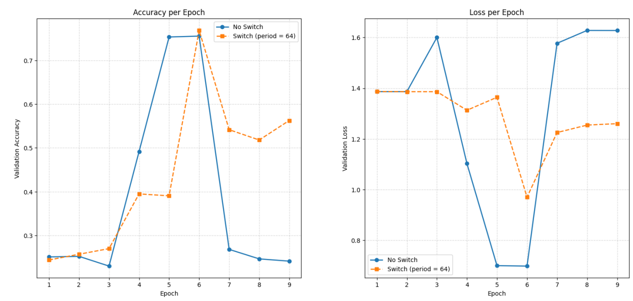


Fig. 2: Validation accuracy and loss across epochs for Main (Model 3)

C. Experiment 1: Context switching sweep for Main (Model 3)

a) Findings: The results in Table IV reveal a strong dependence on switching frequency. With no context switching, the model struggles to retain rule information, achieving only 23% accuracy and a scattered confusion matrix. Introducing moderate switching ($p = 32$) yields a dramatic improvement to 76.8% accuracy, indicating that frequent exposure to new contexts enhances adaptability and stabilizes rule encoding. However, increasing the interval further ($p = 64$ or $p = 128$) degrades performance again, suggesting that overly long static phases cause the model to overfit to specific contexts before the next switch. These findings support the view that context switching acts as an implicit curriculum signal, encouraging flexible abstraction of rule changes rather than rote memorization.

TABLE IV: Experiment 1: Context switching sweep for Main (Model 3).

Switch Period	Test Loss	Test Accuracy	Observation
None	1.6357	0.2305	No rule adaptation, low accuracy
32	**0.9878**	**0.7685**	Stable learning, best performance
64	1.3805	0.3050	Moderate switching, partial collapse
128	1.4052	0.3198	Infrequent switching, weak retention

D. Experiment 2: Supervision Granularity

a) Findings:: Reducing supervision to only the final SEP (corresponding to the query phase) resulted in an 8 percentage-point drop in accuracy compared to full supervision. This suggests that intermediate SEP tokens contribute auxiliary gradient signals that help the model stabilize training and better infer the latent rule context. The equivalence of "last" and "query" supervision confirms that the final SEP aligns precisely with the query segment, validating our dataset design. Overall, full-sequence supervision ("all") remains the most effective approach for generalization.

TABLE V: Experiment 2: Supervision granularity ablation (Main (Model 3), switch period = 64).

Supervision	Test Loss	Test Accuracy
All	1.3805	**0.3050**
Last/Query	1.3864	0.2460

E. Experiment 3: Model Capacity Scaling

Scaling up the transformer beyond the base configuration led to diminishing and even negative returns. While a small model underfit slightly, increasing hidden size and attention heads caused the model to over-specialize and destabilize optimization, reflected in higher loss and lower accuracy. This indicates that the WCST dataset's task complexity does not demand deep or wide transformers; the Base (128-4-4) model offers an optimal trade-off between capacity and generalization. In short: bigger does not mean better, at least not for this reasoning regime.

TABLE VI: Experiment 3: Model capacity scaling for Main (Model 3) (switch period = 64, supervise = all).

Model	d_model	Layers	Heads	Test Loss	Test Accuracy
Small	64	2	2	1.3838	0.2822
Base	128	4	4	**1.3805**	**0.3050**.
Medium	256	4	4	1.5251	0.2187
Large	256	6	8	1.3865	0.2512

F. Experiment 4: Different Train_batch sizes

Increasing the number of training batches per epoch produced only marginal improvements until a threshold of approximately 2000 batches, where accuracy rose from 25 % to 30 %. Smaller training sets (250–1000 batches) yielded near-random performance, suggesting that the model requires extensive exposure to rule transitions to generalize effectively.

This behaviour highlights a data-efficiency bottleneck: the model's inductive bias depends on seeing sufficient context-switch diversity to infer underlying task structure. Once this threshold is reached, the confusion matrix begins to show diagonal dominance, indicating that consistent category learning has emerged.

TABLE VII: Experiment 4: Effect of training set size per epoch (switch period $= 64$, supervise=`all`).

Train Batches	Test Loss	Accuracy
250	1.3869	0.252
500	1.3862	0.252
1000	1.3866	0.253
2000	**1.3805**	**0.3050**

V. Discussion

Across all experiments, our findings reveal that context switching frequency, supervision granularity, model capacity, and dataset scale interact to shape rule-learning performance. The model performs best under frequent but not excessive context changes (period = 32) and full-sequence supervision, which collectively encourage flexible rule abstraction. When rules remain static, the network quickly overfits, while infrequent switching leads to catastrophic forgetting between phases.

Capacity scaling highlights an important asymmetry: increasing the model's width or depth degrades performance, suggesting that over parametrization destabilizes training in small-data reasoning tasks. The optimal configuration (d = 128, L = 4, H = 4) strikes a balance between representational power and gradient stability. Similarly, experiments varying the training batch count show that meaningful generalization emerges only after a critical exposure threshold (2000 batches). Below that, models fail to form coherent category boundaries, pointing to the need for extensive rule-switch diversity to build stable internal abstractions.

Overall, the experiments converge on a central insight: in-context rule inference requires both structured supervision and frequent environmental perturbation. Static regimes or limited exposure suppress the emergence of flexible representations, while excessive scaling or smoothing introduces instability. Though the achieved accuracies remain modest (30%), the results demonstrate that even compact transformers can exhibit the early stages of rule induction under carefully tuned conditions.

VI. Limitations

Several constraints remain:

(i) The WCST environment is a simplified symbolic simulation that may not fully capture the complexity of human cognitive flexibility.

(ii) Training remains unstable across random seeds, implying high sensitivity to initialization.

(iii) Evaluation is limited to a single synthetic generator; broader testing on varied rule structures would strengthen claims of generalization.

(iv) Accuracy metrics alone may understate emergent reasoning; qualitative inspection of attention maps or latent clustering could provide deeper insight.

(v) Although the model achieved its peak accuracy under a switch period of 32, this result proved inconsistent across reruns, with later trials showing substantially lower performance. This variance highlights the stochastic nature of the setup and suggests that the apparent peak may reflect random initialization or transient training dynamics rather than a robust optimum

VII. Data Integrity and Leakage Control

To prevent data leakage, all training, validation, and test streams were generated independently via distinct WCST instances with disjoint random seeds. Each dataset instance reinitializes the rule generator and randomizes category mappings. No sequence or rule context overlaps between splits. All hyper-parameters and configurations are logged to JSON for reproducibility.

VIII. Conclusion

This project demonstrates that a compact Transformer, when trained on a synthetic rule-switching environment, can exhibit basic forms of adaptive reasoning analogous to in-context learning. By systematically varying supervision scope, context-switching frequency, capacity, and data scale, we show that robust rule abstraction emerges only under frequent rule changes and dense supervision.

While absolute performance remains far below human flexibility, these results underscore how architectural simplicity and training dynamics jointly influence adaptive behaviour. Future work could extend this foundation with curriculum learning, larger datasets, or hybrid symbolic–neural systems to approach more general forms of abstract reasoning.

IX. Collaboration

All group members contributed equally to the completion of this Project. The workload was evenly distributed among the three members, with each person participating in data preprocessing, model training, result analysis, and report preparation. Collaboration was maintained through regular discussion and joint review of the code and write-up to ensure accuracy and clarity.

Dr. Devon Jarvis provided the WCST.py file to generate our data.

APPENDIX

NEURIPS PAPER CHECKLIST

1) **Claims**
Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
Answer: [Yes]
Justification: The claims made in the abstract can be found in the methodology and results.
Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2) **Limitations**
Question: Does the paper discuss the limitations of the work performed by the authors?
Answer: [Yes]
Justification: The paper discusses the limitations of training the model.
Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3) **Theory Assumptions and Proofs**
Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
Answer: [NA]
Justification: There are no stated theorems or assumptions in the paper.
Guidelines:
   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.
4) **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
Answer: [Yes]
Justification: The paper provides all needed information to reproduce the main experimental results in the methodology
Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5) **Open access to data and code**
Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
Answer: [Yes]
Justification: The code has been provided in the hand-in.
Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6) **Experimental Setting/Details**
Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?
Answer:[Yes]
Justification: The paper specifies dataset splits (train 2000 / val 300 / test 300), supervision protocol, optimizer (AdamW), and key hyperparameters/controls; it also fixes seeds and provides exact reproduction commands. See Methodology (data

regimes and splits), (loss/optimization), Results (AdamW lr=310^{-4}, betas, weight decay)
Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7) **Experiment Statistical Significance**
Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
Answer: [No]
Justification: Results are reported as point accuracies/tables and confusion matrices without error bars or confidence intervals.
Guidelines:
    - The answer NA means that the paper does not include experiments.
    - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
    - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
    - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
    - The assumptions made should be given (e.g., Normally distributed errors).
    - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
    - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
    - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
    - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8) **Experiments Compute Resources**
Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
Answer: [No]
Justification: These experiments are small enough in scale that any modern pc with a graphics card should be able to run them in a couple minutes.
Guidelines:
    - The answer NA means that the paper does not include experiments.
    - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
    - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
    - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).
9) **Code Of Ethics**
Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?
Answer: [Yes]
Justification: The research in this paper confirms with the NeurIPS Code of Ethics.
Guidelines:
    - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
    - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
    - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).
10) **Broader Impacts**
Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
Answer: [No]

Justification: While technical limitations are discussed, the paper does not analyse positive/negative societal impacts, misuse, fairness, privacy, or security.
Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11) **Safeguards**
Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?
Answer: [No]
Justification: No high-risk models or scraped datasets are released; experiments use a synthetic WCST generator with split isolation and internal logging, so safeguard policies are not applicable here.
Guidelines:
    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12) **Licenses for existing assets**
Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
Answer: [Yes]
Justification: Dr. Devon Jarvis provided the WCST.py code to generate data and is credited in the paper under collaborations.
Guidelines:
    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13) **New Assets**
Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
Answer: [NA]

Justification: The work does not announce a released dataset/model/package; it describes a synthetic set-up and internal logs, with no linked artefact or documentation package. See Reproducibility and Data Integrity

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14) **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [No]

Justification: No human subjects or crowd workers are involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15) **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: As there are no human subjects or crowdsourcing components, IRB considerations do not apply and are not discussed.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.