Forecasting machine degradation of GPU Clusters

Nina Cai
Google Canada
Kitchener, ON
ninacai@google.com

Shuxin Nie
Google
Mountain View, CA
shnie@google.com

Zhehui Chen
Google
Sunnyvale, CA
zhehuichen@google.com

Nupur Gulalkari
Google
Sunnyvale, CA
nupurgulalkari@google.com

George Vanica
Google
Kirkland, WA
gvanica@google.com

Chetna Jain
Google
Kirkland, WA
chetnajain@google.com

Newfel Harrat
Google
Sunnyvale, CA
newfel@google.com

Sethu Sankaran
Google
NYC, NY
sethusankaran@google.com

Abstract

Large-scale training jobs, especially those utilizing GPU clusters, are vulnerable to various failure modes, including individual hardware faults, network issues, and software-level problems. These failures can lead to significant downtime, wasted computational resources, and delays in research or production workflows. We propose a ML based forecasting algorithm designed for predicting health status of GPU clusters. Through extensive ablation studies, we found that cascading 1D CNNs achieved the best performance. The model leverages time-series data representing various cluster metrics, such as temperature, power consumption, and resource utilization towards predicting cluster failures, enabling proactive maintenance and resource optimization. By tuning differently per use-case, the 5-hour forecasting model is able to achieve overall PRAUC of 0.90. This work is motivated by the need to improve the reliability and efficiency of large-scale training jobs that are susceptible to hardware and software failures.

1 Introduction

Large-scale machine learning training jobs, especially those dependent on extensive GPU clusters, represent a significant investment of resources[1-3]. However, this investment is perpetually at risk[4] due to a variety of failure vectors, including hardware malfunctions, network instability, and software-induced errors. These failures translate directly into substantial costs: lost compute cycles, wasted energy consumption, and ultimately, delays in delivering research outcomes or production-ready models. Consequently, achieving dependable and efficient training requires a system engineered for high fault tolerance and designed to rapidly recover from disruptions[5]. Currently, many of these failures are detected reactively[6-8], often after a job has already been disrupted. This reactive approach leads to several challenges such as lost progress due to failures, higher operational costs, difficulty in bug fixing, and a negative user experience.

Therefore, a proactive approach to predicting cluster health is crucial. By developing a machine-learning model capable of identifying clusters at risk of failure, we can trigger preemptive maintenance actions before failures occur, minimizing downtime and maximizing resource utilization. Besides, we could dynamically adjust resource allocation based on predicted health states, ensuring optimal performance and efficiency. We could also identify potential failure patterns earlier, facilitating faster and more effective troubleshooting. By predicting failures, we can improve the overall reliability of the clusters and reduce the impact of failures in the users' workflows. Thus, we could reduce the incidence of job failures, leading to a more reliable and predictable user experience.

This work contributes to these efforts by developing a robust and accurate ML model that can effectively predict cluster health, thereby paving the way for a more reliable and efficient large-scale training infrastructure. The use of focal loss addresses the class imbalance in the data, a common problem in this type of predictive task, and the evaluation by categories provides a more detailed understanding of the model's performance. This will allow a more precise model deployment depending on the specific use case.

2 The Predictor Framework

2.1 Data Acquisition and Feature Engineering

Our proposed model operates on a comprehensive set of time-series data[9], representing various cluster metrics, to assess and forecast cluster health status. The system employs a sliding-window approach, monitoring cluster behavior over a defined *observation window* and predicting its health status for a subsequent *forecasting window*. Specifically, the system captures features over a 3-hour observation window. The system aims to predict machine health for the subsequent 5-hour forecasting window. Feature data is sampled at 10-minute intervals, resulting in 18 discrete snapshots of the cluster's state within the 3-hour observation window. We have performed forecasting on 5-hour window because that is the most valuable across different use cases such as scheduling jobs and elastic training.

The input feature set comprises a diverse collection of hardware and network metrics. Hardware metrics include: (i) Temperature, representing measured temperature readings, (ii) Tlimit, representing the maximum allowable temperature, (iii) SM Utilization, denoting the utilization of Streaming Multiprocessors, and (iv) Contamination Ratio, reflecting the level of contamination within the cluster. Network metrics include (i) *throughput rx bytes delta* and *throughput tx bytes delta*, representing the change in bytes received and transmitted, respectively, (ii) *throughput rx bytes count* and *throughput tx bytes count*, representing the total received and transmitted bytes, respectively, (iii) *packets retransmission delta* and *packets retransmission count*, representing the change in retransmitted packets and the total retransmitted packets, respectively, (iv) *goodput rx bytes delta* and *goodput tx bytes delta*, representing the change in goodput received and transmitted bytes, respectively, and (v) *goodput rx bytes count* and *goodput tx bytes count*, representing total goodput received and transmitted bytes respectively. In addition, we consider *inter fabric bytes*, *intra tor bytes*, *intra fabric bytes* and *intra superblock bytes*, representing the bytes transmitted between fabrics, within the top-of-rack switch, within the fabric and within the superblock respectively. Each metric is captured at discrete 10-minute intervals, forming a sequence of measurements, or snapshots, for each machine. Finally, a categorical feature, Health Status, reflecting the machine's current health state, is also included as part of the input features.

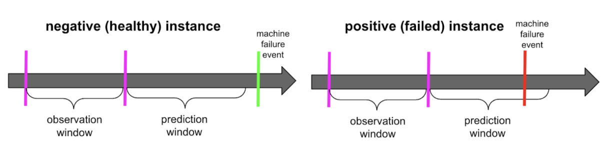

Figure 1: Illustration of the observation window.

2.2 Model Architecture

Our proposed system employs a dual-model architecture, consisting of two distinct 1D Convolutional Neural Network (1D-CNN) models, designed to capture different aspects of the anomaly detection problem. These models, referred to as Hard Sample Detection (Model 1) and Easy Sample Detection (Model 2), are trained separately and leveraged in a cascading manner.

Both models share a similar architecture, beginning with an Input Layer that accepts a sequence of features, represented as a tensor with dimensions (number of snapshots, number of features). The input data then passes through three sets of Convolutional Layers, each comprising a 1D convolution (Conv1D) followed by a MaxPooling1D layer. The Conv1D layers utilize Rectified Linear Unit (ReLU) activation functions to introduce non-linearity and learn local temporal patterns within the time-series data. The number of filters for each Conv1D layer is set to 64, 32, and 32, respectively, for Model 1, and the same values for Model 2. MaxPooling layers are used to downsample the data, reducing the dimensionality and computational complexity. The specific kernel sizes are configurable. Following the convolutional layers, the output is flattened by a Flatten Layer into a single vector. This vector is then passed through a Dense Layer, also with ReLU activation, for further feature transformation. To mitigate overfitting, a Dropout Layer with a rate of 0.5 is applied after the dense layer. Finally, a Dense Output Layer, with a single neuron and a Sigmoid activation function, generates a probability score in the range of [0, 1], representing the likelihood of the cluster transitioning to an unhealthy state.

To address the class imbalance commonly observed in this type of predictive task, we incorporate a Focal Loss function during model training. This loss function adjusts the weights of the samples based on their classification difficulty, providing greater emphasis to misclassified samples.

Model 1, the Hard Sample Detection model, is trained using time-series data of GPU and network metrics from samples where the labels of the observation window and the prediction window are different, including both samples with clean and contaminated data. This model is designed to be robust to cases where the cluster exhibits unstable or rapidly changing behavior.

Model 2, the Easy Sample Detection model, is trained using time-series data of GPU and network metrics, including the contamination ratio as an extra input feature, from samples where the labels of the observation window and the prediction window are the same, and the data is considered to be more clear, using only samples with clean data. This model is employed for samples where the cluster's behavior is more consistent and well-defined.

The outputs from both models are combined using a cascading logic strategy, which is detailed in the section below.

2.3 Cascading Logic

We propose a two-stage anomaly detection framework leveraging a cascading model architecture. Initially, the input data is processed by Model 1, a Hard Sample Detection model, which generates a probabilistic score for each sample. This score reflects the model's confidence in classifying a sample's health status. Samples are then categorized based on this confidence: those with scores in the range [0, 0.3] are considered healthy, while those in (0.3, 1] are deemed unhealthy. Furthermore, samples are subjected to a contamination ratio check. Specifically, samples that Model 1 labels as unhealthy and where the observation window contamination ratio is 0, or those labeled as healthy with a contamination ratio > 0, retain the probability generated by Model 1. The remaining samples, for which Model 1 exhibits lower confidence, are subsequently processed by Model 2, an Easy Sample Detection model. This model generates a probability score for these samples. Finally, the anomaly prediction is determined by combining the probability scores from Model 1 and Model 2 following a cascading strategy. The output of this combined prediction is then classified as healthy if the final probability score is in the range [0, 0.5) or as unhealthy otherwise. Additionally, we provide a default categorization of the final result to users: [0, 0.4) is classified as healthy, [0.4, 0.5) as at risk, and (0.5, 1] as unhealthy.

2.4 Output Threshold Selection

Following model predictions, we establish a framework for output thresholding and performance evaluation. Both Model 1 and Model 2 produce a continuous probability score (ranging from 0 to 1), reflecting the likelihood of a sample being unhealthy; a higher score indicates a higher likelihood. For evaluation, we apply a binary classification (healthy/unhealthy) using a threshold of 0.5, though this can be adjusted based on specific application needs. For user feedback, we provide a categorical prediction using thresholds of 0.4 and 0.5, with scores in [0, 0.4) classified as "HEALTHY," [0.4, 0.5) as "AT RISK," and (0.5, 1] as "UNHEALTHY." To assess model performance across different data subsets, we conduct a stratified evaluation based on the observation window's contamination ratio.

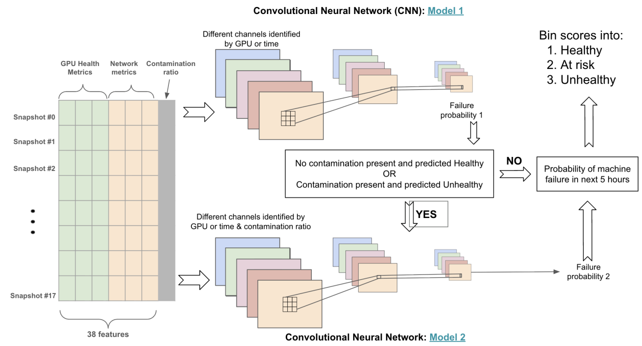

Figure 2: Model Training Pipeline including Cascading Logic.

Furthermore, we utilize Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves for a comprehensive analysis, enabling the selection of an optimal probability threshold. Finally, the training process incorporates a focal loss function with tunable hyperparameters α and γ to influence the importance of samples during training.

3 Experiment

A comprehensive dataset spanning 180 days of time-series data, sampled at 10-minute intervals, was assembled for this study. This dataset encompasses approximately 1.8 million training data points and 610,000 testing data points, representing 13,000 and 4,000 distinct machines, respectively. To ensure generalizability of the results, the dataset was partitioned by machine name into mutually exclusive training (60%), validation (20%), and testing (20%) sets. The feature preprocessing pipeline, implemented using Flume, was designed for efficient processing, completing within a few hours.

All model trainings were conducted on a NVIDIA H100 and H200 GPU accelerator instance.

The model's performance is evaluated by different categories, based on if there are any unhealthy signals during the observation window. We mainly focus on two categories:

(1) Category 1: Clean Set: Positive (Clean) + Negative (Clean): Only Healthy in the observation window and Healthy/Unhealthy in the Prediction Window

(2) Category 2: Full Set: Healthy, Unhealthy in the Observation Window, Healthy/Unhealthy in Prediction Window

We compute standard metrics like Precision, Recall, Accuracy, PR-AUC and AU-ROC for two categories for the Cascading Model.

	Precision	Recall	F1-score
Healthy	1.00	0.76	0.86
Unhealthy	0.15	0.93	0.26
Accuracy	0.767659		
AUROC	0.945526		
PRAUC	0.901844		

Table 1: Performance Metrics for Anomaly Detection

It is important to note that the proposed model architecture is specifically designed for and validated on NVIDIA H100 and H200 GPU accelerator platforms. The model's performance and applicability on other hardware configurations have not been evaluated.

References

[1] J. S. Vetter, R. Glassbrook et al., *Keeneland: Bringing heterogeneous GPU computing to the computational science community.* Computing in Science & Engineering, vol. 13, no. 5, pp. 90–95, 2011.

[2] D. Kothe and R. Kendall, *Computational science requirements for leadership computing.* Oak Ridge National Laboratory, Technical Report, 2007.

[3] C. L. Mendes, B. Bode et al., *Deploying a large petascale system: The blue waters experience.* Procedia Computer Science, vol. 29, pp. 198–209, 2014.

[4] S. K. S. Hari, T. Tsai et al., *SASSIFI: Evaluating resilience of GPU applications.* in Proceedings of the Workshop on Silicon Errors in Logic-System Effects (SELSE), 2015.

[5] D. Tiwari, S. Gupta et al., *Understanding GPU errors on large-scale HPC systems and the implications for system design and operation.* in High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on. IEEE, 2015, pp. 331–342.

[6] *Understanding XID errors* `http://docs.nvidia.com/deploy/xid-errors/index.html`, 2015

[7] *Validate the cluster level NCCL test with 4 nodes and 32 GPUs* `https://docs.nvidia.com/dgx-basepod/deployment-guide-dgx-basepod/latest/mn-nccl.html`, 2024-2025

[8] *NVIDIA DCGM* `https://developer.nvidia.com/dcgm`, 2025

[9] G. E. Box, G. M. Jenkins et al., *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

A Supplementary Experiment Results

This section provides supplementary ablation studies to further analyze the behavior and performance of our proposed model. We investigate the impact of different model architectures and variations in time horizon configurations.

Besides PRAUC and AUROC, additional metrics like False Rejection Rate (FRR) and False Acceptance Rate (FAR) are computed for each category for the five-hour Cascading Model.

- FRR at 4 FAR thresholds - 0.1, 0.2, 0.3, 0.4
- FAR at 3 FRR thresholds - 0.1, 0.05, 0.01

	Category 1	Category 2
FRR@FAR=0.1	0.688078	0.001463
FRR@FAR=0.2	0.458821	0.000000
FRR@FAR=0.3	0.299794	0.000000
FRR@FAR=0.4	0.193946	0.000000
FAR@FRR=0.1	0.582722	0.071996
FAR@FRR=0.05	0.744777	0.082865
FAR@FRR=0.01	0.941276	0.096057

Table 2: FRR@FAR and FAR@FRR

A.1 Time Horizon Ablations

To understand the effect of the forecasting horizon on model performance, we conducted an ablation study by varying the forecasting window from 0.5 hours to 24 hours, while keeping the observation window fixed. The results, presented in Table 3, demonstrate distinct trends for Category 2.

Category 2 shows high predictive performance for shorter horizons, which degrades as the forecasting window increases. Both PRAUC and AUROC for Category 2 start at 0.97-0.98 for windows of 0.5-1 hour and steadily decrease to 0.81 (PRAUC) and 0.91 (AUROC) at the 24-hour mark. This indicates that predicting Category 2 outcomes becomes substantially more difficult over longer time frames.

The choice of a 5-hour forecasting window, highlighted in the table, represents a compromise maintaining strong performance (0.90 PRAUC and 0.95 AUROC) for Category 2. The optimal window length may depend on the specific application and the relative importance of detecting each category.

Fcst. Window	Metric	Value
0.5 hours	PRAUC	0.970
	AUROC	0.970
1 hour	PRAUC	0.980
	AUROC	0.980
3 hours	PRAUC	0.940
	AUROC	0.960
5 hours	PRAUC	**0.900**
	AUROC	**0.950**
10 hours	PRAUC	0.890
	AUROC	0.940
15 hours	PRAUC	0.870
	AUROC	0.940
24 hours	PRAUC	0.810
	AUROC	0.910

Table 3: Time Horizon Ablation Results. Impact of varying forecasting window lengths on model performance on the full set.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract and introduction reflects the paper's contributions and scope accurately.

   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 the work performed by the authors.

   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 is no theroretical result.

   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 fully discloses all the information needed to reproduce main experimental results.

   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: [No]

   Justification: The paper doesn't provide open access to the data and code.

   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 all the training and test details.

   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: [Yes]

   Justification: The paper reports error bars suitably and correctly.

   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: [Yes]

   Justification: The paper provides sufficient information on the computer resources.

   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 conducted in the paper conforms 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: [NA]

    Justification: There is no societal impact of the work performed.

    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: [NA]

    Justification: The paper poses no risks.

    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: The paper provides the patent information.

    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 paper does not release new assets.

    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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.