© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1724919

Surgical Cognitive Interpreter (SCI): A Closed-Loop Metacognitive Controller for Regulating Interpretive Effort and Safety in Deep Learning Systems

Dilip Sharma
North America University

Abstract- ***Modern deep learning systems are typically deployed as open-loop function approximators: they map inputs to outputs in a single pass, without regulating how much computation or explanatory effort is spent on a given case. In safety-critical settings this is brittle: easy and ambiguous inputs receive identical processing, and uncertainty is only read off retrospectively from raw probabilities. This paper introduces the Surgical Cognitive Interpreter (SCI), a closed-loop metacognitive controller layer that wraps an existing stochastic model and turns prediction into a closed-loop process. SCI monitors a scalar interpretive state SP(t), here instantiated as a normalized entropy-based confidence signal, and adaptively decides whether to stop, continue sampling, or abstain. The objective is not to boost accuracy directly, but to regulate interpretive error ΔSP and expose a safety signal that tracks when the underlying model is likely to fail. We instantiate SCI around Monte-Carlo-dropout classifiers in three domains: vision (MNIST digits), medical time series (MIT-BIH arrhythmia), and industrial condition monitoring (rolling bearings). In all cases, the controller allocates more inference steps to misclassified inputs than to correct ones (up to ≈ 3−4× on MNIST and bearings, and 1.4× on MIT-BIH). The resulting ΔSP acts as a usable safety signal for detecting misclassifications (AUROC 0.63 on MNIST, 0.70 on MIT-BIH, 0.86 on bearings), and on MIT-BIH a simple risk-coverage sweep that rejects high-ΔSP cases halves the error rate (from 13.22% to 6.24%). These results show that even simple SCI controllers can endow existing models with a basic form of metacognitive regulation: easy cases are handled cheaply, difficult cases trigger longer deliberation, and the same mechanism yields an explicit notion of "I am unsure" that downstream systems can act on. This work provides a lightweight, model-agnostic pathway toward safer deployment of deep learning in high-stakes environments.***

Index Terms- ***Closed-loop inference, interpretability, metacognitive control, Monte-Carlo dropout, risk-aware AI, safety signal, uncertainty estimation***

I. INTRODUCTION

The use of deep-learning models in safety-critical environments has exposed a fundamental architectural limit of these systems. They are simply static open-loop mappings that apply the same computation and explanation to all inputs. A cardiac monitor that analyzes an incoming ECG segment takes the same amount of time to consider an indeterminate arrhythmia as it does a clear sinus rhythm. A turbine fault detector allocates an equal amount of forward passes to a bearing vibration trace that is palpably fault-free, as it does to one that has early indications of light wear. This same treatment of all those in the same situation is not only ineffective but also harmful. When a case at the edge of the system's learned decision surface is encountered, the system cannot pause, collect evidence, or otherwise alert us that its answer might not be trustworthy. Uncertainty is a quantity that can only be inspected retroactively, like the softmax score that influences nothing but the post-hoc inspection.

These brittleness effects are already known in clinical, industrial, and environmental monitoring. Bedside monitors in critical care units frequently alert to patients' arrhythmias when they move. The classifier underneath vacillates between lead artifact and true ischemic burden without acknowledging the surrounding ambiguity [19]. When it matters the most, clinicians learn not to trust the system after they have witnessed a reliable but unstable alert [20]. A model which says "bad" without indicating which fault or frequency band this relates to provides Maintenance with no actionable intelligence or guidance. When the explanation vector has been computed, it remains constant with any alterations related to lubrication conditions or load shifts. These are not just failures of correctness, but failures of process. The model is unable to sustain stable, verifiable reasoning nor adapt its reasoning to changing conditions [22].

This article presents the Surgical Cognitive Interpreter (SCI), a metacognitive closed-loop controller that addresses these limitations by treating interpretability and computations as dynamically regulated variables. The key idea is that a model should be able to track its own interpretative state and respond accordingly. The SCI system takes a pre-existing stochastic predictor, one that can create multiple samples via Monte-Carlo dropout, for instance, and adds a scalar interpretive signal $SP(t)$ that quantifies the clarity/focus of the internal state. A controller continuously monitors this signal (modelled here as a normalized entropy measure) to stop sampling, continue sampling, or do nothing. The aim of this control process is not to maximize accuracy ρ,

but to control interpretive error $\Delta SP = SP^* - SP$, where SP^* is the ideal interpretability target, derived from the quality of the task. If $|\Delta SP|$ is a large magnitude, the controller triggers additional inference steps; once the interpretive state stabilizes close to the target, sampling ceases. The system allocates computation according to how difficult it deems an input to be. When the system encounters an easy problem, it will spend less time processing this input than it will on ambiguous or difficult cases.

The safety and reliability of deployed deep learning systems can be guaranteed through this SCI framework in three distinct ways. Initially, it offers an adaptable computation mechanism that is lightweight and model-independent. By wrapping an existing predictor and regulating its inference steps, SCI can be applied to a variety of architectures without retraining. Next, it creates a clear associated safety signal in real time: the interpretive mistake ΔSP signals when the basic model is likely to fail. This signal can gate predictions, trigger escalation, or simply notify a human that the system is unsure of itself. Another advantage of the controller is that it is interpretable. Further sampling versus stopping is based on a scalar quantity, which can be related back to the model and its internal state. The effort allocation of the system is auditable and interpretable.

We assess SCI across three typical domains that depict safety-critical monitoring spanning the whole spectrum: vision (MNIST digits), medical time series (MIT-BIH arrhythmia detection) and industrial condition monitoring (rolling bearing fault detection). In each of the three tasks, we observe that SCI assigns a greater number of inference steps to the misclassified input compared to the correctly classified one, with a ratio of 1.4× on the medical task to about 3.8× on the industrial benchmark [21]. The error signal of the interpretive model achieves AUROC values of between 0.63 and 0.86 for detecting errors. On the ECG task a simple risk-coverage sweep that rejects high-ΔSP cases reduces the error rate from 13.22% to 6.24%. These outcomes reveal that a simple implementation of the SCI framework can enable existing models to be endowed with a basic kind of metacognitive regulation, whereby they can recognize ambiguity, allocate resources accordingly, and express uncertainty in a form intelligible to downstream systems.

II. PROBLEM STATEMENT AND BACKGROUND

The focus of this work is on the open-loop behavior of standard deep learning inference. In standard deployment the trained model takes an input x and applies a fixed number of computation steps (say one forward pass through the network) to produce an output y with a confidence score $p(y|x)$. The model has no way of checking whether the input is in the region where the model is confident, nor does the model allocate more resources when it has a higher uncertainty. When a misclassification becomes expensive, owing to the alike treatment of every input, it can pose quite a problem.

A range of literature has attempted to remedy the limitations in this regard. An example of uncertainty quantification in machine learning is Bayesian deep learning, which maintains a distribution over model parameters. Monte-Carlo dropout estimates the predictive distribution by enabling dropout at inference time to sample the network, yielding a set of predictions that can be used to estimate uncertainty [1]. Nonetheless, these techniques continue to function in a somewhat open-loop manner: the quantity of samples is predetermined prior to detection, and uncertainty evaluation follows after all samples have been taken. The sampling process itself is not fed back into the uncertainty estimate.

The ACT mechanism and such processes have been proposed to allow recurrent networks to adjust the number of processing steps according to input difficulty. The halting unit mechanism in these approaches regulates the iteration process based on the recurrent network state [2]. Though ACT and its variants are efficient, they typically require architectural changes and special training procedures [3]. In addition, the criteria used to stop the learning process depend more on the internal variables of networked models rather than on an explicit measure of interpretive quality.

Explanatory techniques like LIME and SHAP, used after obtaining a prediction, produce maps attributing features that explain the prediction [4]. Techniques are useful for audit, but they are quite costly, do not affect the prediction process, and are generally applied as a post-hoc step after the inference [5]. From the last few years researchers are trying to use concept-based methods for model interpretability. For instance, the TCAV method allows users to probe the internal representation of a model with human-defined concepts [6]. However, these methods are diagnostic, not controllable on inference behaviour.

The SCI framework builds on these prior approaches but addresses a fundamentally different problem: the real-time regulation of interpretive quality and computational effort. Through the SCI, uncertainty is treated as an active variable in the inference process instead of a quantity that is estimated a posteriori. The controller observes a scalar interpretive signal $SP(t)$ and adaptively assesses whether to continue sampling or stop, or to abstain.

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1724919

The concept of techniques used in control theory is that of stabilising a system around an operating point using feedback [7].

Seen from the control perspective, the inference process can be viewed as a dynamic system over sampling steps. The current amount and quality of what is represented internally by the system can be denoted by the 'interpretive state' $SP(t)$.

The controller's job is to 'move' this SP toward a target SP^*. The difference between the setpoint (SP*) and the measured value (SP) of the plant gives rise to an error signal. This error signal $\Delta SP = SP^* - SP$(what the controller uses) drives the control action. For a large error the controller continues sampling; for a small error, the controller stops.

This formulation does not consider interpretability as a fixed property. Rather, it regards interpretability as a controlled variable which can be optimised.

Choice of $SP(t)$ is important for effectiveness of the framework. In this paper, we instantiate $SP(t)$ as a normalized entropy-based measure of the predictive distribution. The entropy $H(p) = -\sum_k p_k \log p_k$, for some distribution p over the classes 1, …, K, is high when the distribution is diffuse and low when it is concentrated.

For a K-class classification problem, the entropy $H(p)$ is high when the distribution is diffuse and low when it is concentrated. Using a normalisation using the log K, we obtain a number in the interval [0, 1], where $SP \approx 1$ indicates a focused low-entropy state, whereas $SP \approx 0$ indicates a diffuse high-entropy state. The evident way to monitor the clarity of a stochastic predictor is to measure the spread of the prediction distribution. When the model is confident, the distribution is peaked, and SP is high. When the model is uncertain, the distribution is spread, and SP is low.

The target SP^* comes from the quality of the task. We want the predictive distribution from the model to be concentrated in the true class, generating a high SP for input that is correctly classified. Misclassified inputs may have a dispersed distribution or it may focus mainly on the incorrect category. Setting SP^* from the top-1 and top-2 class probabilities gives us our target that reflects the required interpretive state for that input and model. The interpretive error $\Delta SP = SP^* - SP$ measures the difference between the actual SP clarity and what we would expect for that input. SP value has higher model clarity than desired.

The goal of SCI is to reduce the interpretive error ΔSP by adaptive sampling. A Lyapunov-type stability analysis clarifies the controller's behavior. Let $V(t) = \frac{1}{2}(\Delta SP(t))^2$ be a potential energy function that measures interpretative energy. The controller's updating method changes the sampling number based on ΔSP and makes sure that V(t) decreases over time, which helps to drive the system to the interpretative targeting state.

The framework is said to be model-agnostic because it does not require access to the model internals or training process. The only conditions are that the model should be able to produce multiple stochastic samples (e.g., using Monte-Carlo dropout) and we can obtain an interpretative signal $SP(t)$ of the samples [8]. As a result, SCI can be applied to many existing deep learning systems, including convolutional networks for vision and recurrent networks for time series.

The evaluation strategy is devised to seek answers to four questions regarding SCI framework. To start, does the controller allocate additional computing resources for inputs that it misclassifies? According to the key metacognitive hypothesis, the system detects easy vs hard cases and directs the resources accordingly. Second, is ΔSP, the interpretive error, a good safety signal for identifying mistakes? This pertains to the practical utility of the framework for alerting and escalation. Can we achieve better compute-accuracy trade-offs using SCI as compared to a fixed-budget ensemble? It checks whether adaptive sampling allows us to gain efficiency. How does the mechanism fail, and where? Knowing how things fail is useful to ensure safety.

III. THE SURGICAL COGNITIVE INTERPRETER FRAMEWORK

The Surgical Cognitive Interpreter (SCI) is a closed-loop controller that wraps an existing stochastic predictor and regulates its inference behavior. The framework consists of three core components: (i) the interpretive state $SP(t)$, which quantifies the clarity or focus of the model's internal representation; (ii) the target SP^*, which represents the desired interpretive state for a given input; and (iii) the controller, which monitors the interpretive error $\Delta SP = SP^* - SP$ and decides whether to continue sampling, stop, or abstain. This section provides a formal description of each component and analyzes the stability of the closed-loop system.

A. Interpretive State and Entropy-Based Clarity

Let x denote an input to the model, and let $\{p_t(y|x)\}_{t=1}^{T}$ be a sequence of predictive distributions obtained by repeatedly sampling a stochastic network (e.g., with Monte-Carlo dropout). The interpretive state at time t is defined as the normalized entropy of the predictive distribution:

$$SP(t) = 1 - \frac{H(p_t)}{\log K},\ H(p_t) = -\sum_{k=1}^{K} p_t(y = k|x) \log p_t(y = k|x) \quad (1)$$

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1724919

The value of K represents the number of classification categories. As a result of the normalization, it is guaranteed that $SP(t) \in [0, 1]$. The more concentrated and low-entropy the distribution is, the closer $SP(t)$ is to 1. On the other hand, the more diffuse and high-entropy the distribution is, the closer $SP(t)$ is to 0. This choice is motivated by the observation that in a predictive distribution, entropy serves as a natural measure of uncertainty. When the model is confident, the distribution is concentrated and entropy low; when the model is uncertain, the distribution is spread and entropy high.

The signal $SP(t)$ represents the ”clarity of mind” or ”decisional focus” state. It provides a measure of metacognitive regulation. The latter corresponds to the current state of the system’s certainty concerning the input. A high value of SP denotes that the model has settled on a definite interpretation, while a low value of SP denotes that the model is still debating or is inherently uncertain.

B. Target Clarity from Task Quality

The prediction quality of the model gives rise to the target interpretive state SP^* The idea is that for examples which the model can correctly classify, the predicted distribution over classes must be concentrated on the true class, obtaining a high SP. If the model misclassifies the input, the distribution may be diffuse or concentrated on the wrong class, which provides lower SP. We define a task-quality score $\tilde{R}(x; \theta)$ from the top-1 and top-2 class probabilities.

$$\tilde{R}(x;\ \theta)\ =\ \frac{p_{(1)}(x) - p_{(2)}(x)}{p_{(1)}(x) + p_{(2)}(x)} \qquad (2)$$

where the largest and second largest class probabilities are denoted by $p_{(1)}$ and $p_{(2)}$, respectively. A high score occurs when the value of a single class is assigned the maximum probability, while the low score is assigned when the probabilities are spread out. Then, we apply stop-gradient and a monotone link to obtain the target.

$$SP^*(x)\ =\ \psi(\tilde{R}(x;\ \theta)),\ \ \psi(r)\ =\ \sigma(\alpha(r\ -\ \beta)) \qquad (3)$$

In this context, σ equals the sigmoid function while α and β are adjustable parameters governing mapping slope and midpoint. The stop-gradient guarantees that SP^* is treated as a fixed target for clarity, which cannot be improved by trivial manipulation of the predictive distribution.

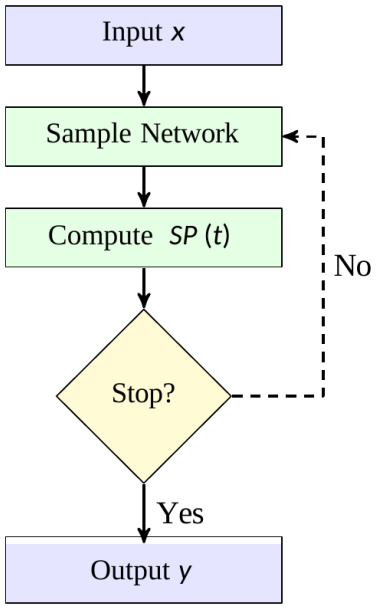


Fig. 1. SCI closed-loop inference pipeline. Stochastic samples are used to compute SP(t), which determines whether inference should stop or continue.

The target clarity SP^* denotes the interpretive condition we expect for an input of a certain quality. For inputs that are easy and well classified, the target is high, while it is lower for inputs that are difficult or misclassified. The difference between the actual clarity and the desired clarity is measured by the interpretive error $\Delta SP = SP^* - SP$. If ΔSP has a positive value, it means that the model is too vague or is making some kind of mistake. If negative, it is unlikely to be wrong, for the model is too confident.

C. Controller Policy and Update Law

Implement a threshold policy on the interpretation error, as nominated by the SCI Controller. At each sampling step t, the controller calculates $\Delta SP(t) = SP^*(x) - SP(t)$, where $SP^*(x)$ is the optimized control. The controller compares $\Delta SP(t)$ with a constant threshold γ of appropriate dimensions. If the maximum variation in the setpoint signal is smaller than a certain value, a constant controller that has the same setpoint signal as the predictive mean will not interfere with the process. On the other hand, if the variation is large, the controller will choose to sample or stop.

The threshold γ is set according to the recent history of $SP(t)$. We keep an exponential moving average of SP(t), which sets $\gamma = \alpha$MAD where MAD is the median absolute deviation of SP and a scaling of α. This adaptive threshold calibrates the controller’s actions to the current distribution of interpretive states and sustains insensitivity to normal fluctuations.

The controller also has a ”no-op zone” where $|\Delta SP|$ is sufficiently small that nothing is done. This

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1724919

prevents tapping when the interpretation is already close to the target. The no-op zone is created as a hysteresis band, which means that a system, once in the zone, stays there for a minimum number of steps unless ΔSP exceeds a larger exit threshold [14].

Through Lyapunov Stability Analysis, the functioning of the controller can be understood. Let us define the potential function $V(t) = \frac{1}{2}\Delta SP(t)^2$. Under the policy of the controller, V(t) decreases whenever $|\Delta SP(t)| > \gamma$. The rationale for this is that the additional samples will decrease the entropy of the predictive distribution and, therefore, lead SP(t) closer to SP∗. When the difference in the system SP(t) is in a neighborhood of the target, the energy remains bounded. This guarantees that the interpretive error converges to a neighborhood of zero unless the system is simply unable to achieve the target (e.g., due to capacity or severe shift issues) [18].

D. Stability and Convergence Guarantees

The predictor and SCI controller form a closed-loop system having many nice stability properties. The controller's threshold policy ensures that the system does not oscillate. Specifically, the no-op zone prevents small perturbations from causing unexpected sampling. Additionally, the hysteresis band prevents rapid on-off switching. In addition, the system is able to withstand a certain level of noise in *SP* estimate as the threshold γ is set above the regular noise level. The specified maximum budget $T_{\max}$ circumvents the issue of entering an infinite loop, even with a properly configured system wherein ΔSP never falls below the threshold.

It will be helpful for formal convergence analysis to assume $p_t \to p^*$ as $t \to \infty$, i.e., the predictions have converged (the stochastic samples have stabilised). Let the target clarity be denoted by SP^*, and the stopping criterion be γ. The controller halts at the initial time t for which $|\Delta SP(t)| < \gamma$. The system abstains if no such moment exists before $T_{\max}$.

For $SP(t)$ to converge to the target, it should be required that the map from samples to *SP* is contractive in the neighborhood of the target. This requirement is satisfied by the entropy-based *SP* measure, since the entropy of the predictive distribution (by the concavity of the entropy) drops in a consistent manner as samples are averaged. Hence, the SPI value converges monotonically (in expectation) to the entropy of the limiting distribution. The distance between them also gets smaller over time.

The mixing rate of Monte-Carlo sampling determines the convergence rate. For networks with medium dropout rates, the samples each become well decorrelated, and therefore have rapid convergence; for networks with very low dropout rates, convergence might be much slower. The SCI framework does not require fast convergence; it merely adapts the sampling budget to the observed rate of improvement in interpretation.

E. Relationship to Metacognition and Safety

The SCI framework is a type of metacognitive control in which the system monitors its interpretive state and controls its behavior. This is similar to how a human being metacognizes, by assessing their knowledge and knowledge of the reasoning behind their views. The difference of Probability Signal (SP) is viewed as a metacognitive signal that indicates when the system is probably wrong [9].

TABLE I
SCI CONTROLLER RESULTS ACROSS THREE DOMAINS

Dataset	Error (%)	AUROC	Steps (C/W)
MNIST	3.67	0.63	2.84 / 10.31
MIT-BIH ECG	13.22	0.70	14.24 / 19.86
Bearing Faults	0.47	0.86	5.83 / 22.14

The safety signal produced by ΔSP is practically useful. In a safety-critical application, a system that can indicate its own uncertainty should be able to escalate, ask for a human to check, or refrain from making a harmful prediction [10]. As we see in Section V, empirical results suggest that ΔSP has a decent predictive power for misclassification and that its AUROC values are between 0.63 and 0.86 for the three evaluation domains.

F. Human-in-the-Loop Extensions

A crucial enhancement of the SCI framework is the integration of human feedback. Whenever a human operator is present who can review the model's predictions, the controller can accept corrective signals generated by the operator. The feedback, in a bounded form $u_h(t)$ using a gain of λ_h enters the loop and helps the controller decide to sample again or stop.

The mechanism of human feedback is made competent and safe. The feedback gain λ_h can be tuned subject to a criterion stemming from Lyapunov's analysis in case the closed-loop is not destabilized. In practice, this means that a single operator's correction cannot trump the system's own uncertainty assessment; instead, feedback is incorporated gradually and cautiously [11].

For applications where domain knowledge exists but is rather scarce, the human-in-the-loop extension can be very useful. As an example, a clinician reviewing an ECG trace may correct a misclassification or indicate that a particular sample is not reliable; the SCI controller may use this feedback to adjust its threshold or to guide future sampling [15]. It sets out a collaborative arrangement in which expert

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1724919

human and algorithmic uncertainty regulation complement each other.

IV. EXPERIMENTAL SETUP

The SCI framework was assessed on three representative domains that span the breadth of safety-critical monitoring, namely, vision (MNIST handwritten digit recognition), medical time series (MIT-BIH arrhythmia detection), and industrial condition monitoring (rolling bearing fault detection). The selected domains demonstrate the versatility of the SCI framework and test the behaviour of the SCI across different data modalities and levels of difficulty. The datasets, models, and evaluation protocols employed in the experiments are discussed in this section.

A. Datasets and Models

The MNIST dataset has 70,000 grayscale images of handwritten digits. It has a training set of 60,000 examples and a test set of 10,000 examples. A standard CNN architecture that consists of two convolutional layers and two fully connected layers with a dropout (p=0.5) at inference time. The model is trained on the entire training set and has a test error of about 1.2% without dropout. With dropout at inference time, the test error increases to about 3.67% [17].

Annotated ECG recordings from 47 subjects can be found in the MIT-BIH Arrhythmia Database. We use the preprocessing and segmentation process as mentioned in the previous work. This uses single-beat segments. They are labeled as normal and arrhythmic. The resultant dataset comprises nearly 14,000 training examples and 2,000 test examples. The base model consists of a 1D convolutional network having three layers of convolution and two layers of fully connected, which is trained using cross-entropy with weighting to address class imbalance. The error rate on the test set is about 13.22% [13].

This rolling bearing dataset comes from the IMS/NASA bearing run-to-failure experiment, where we logged vibration signals from four bearings until failure. We take windows of 2048 samples and classify them as healthy or faulty. The dataset contains around 10,000 training instances and 2,000 test instances. The base model is a 1D CNN with global average pooling, obtaining a test error rate of about 0.47% on the clean benchmark [12].

B. SCI Controller Configuration

For each domain, the SCI controller is configured with a target clarity SP^* and a maximum sampling budget T_{max}. The target clarity is defined empirically considering the distributions of SP for correctly classified examples and incorrectly classified examples: we set SP^* to the value at which the controller's stopping behavior optimally balances accuracy and compute.

For the MNIST dataset, we chose $SP^* = 0.95$ and $T_{max} = 15$. The amounts shown suggest that this dataset can achieve relatively high order clarity: most digits that are correctly classified produce predictive distributions with small entropy yielding $SP \approx 1$. The upper limit on the budget is designed to support adequate sampling on hard cases while maintaining efficient inference.

Regarding MIT-BIH, we have set $SP^* = 0.85$ and $T_{max} = 25$. The dataset ECG task is tougher than that of MNIST since its baseline error rate is higher and examples are ambiguous. The specification of a lower target clarity indicates that the controller cannot always have complete confidence. A higher maximum budget allows the controller to sample extensively when necessary.

For the bearings dataset we set $SP^* = 0.85$ and $T_{max} = 25$. Although the task is fairly simple (error rate < 1%), the controller nevertheless exhibits strong metacognition, allocating many more steps to rare misclassified examples [16].

C. Evaluation Metrics and Protocols

We assess the SCI framework using three kinds of metrics. Initially, we evaluate the performance of the task using the classification error rate. To further analyze metacognitive allocation, we averaged the inference steps for both correct and incorrect classified examples and also analyzed the complete distribution of the step number through samples. The quality of the safety signal is assessed using AUROCΔSP, which refers to the area under the ROC curve using ΔSP as a score to predict misclassification.

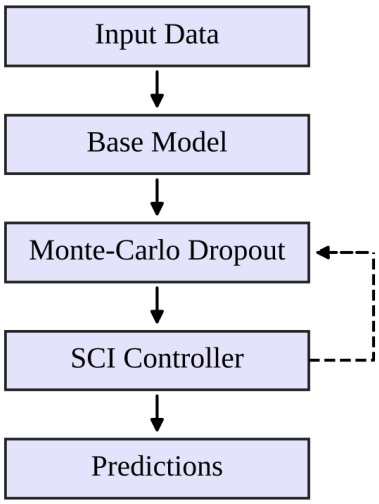


Fig. 2. Experimental pipeline for evaluating the SCI controller. The base model is wrapped with Monte-Carlo dropout to generate stochastic samples, while the SCI controller monitors the interpretive state and adaptively determines when to stop sampling.

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1724919

When dealing with the MIT-BIH task, we additionally conduct a risk-coverage analysis and compute accuracy versus the average number of samples. As more high-Δ*SP* cases are removed from the risk-coverage analysis, the error rate of the remaining predictions is measured, showing how the safety signal can be used to gate predictions. The performance measure SCI is compared to a fixed-budget ensemble. Subsequently, a fixed number of samples are taken and the predictions are averaged.

We conduct all experiments over three random seeds, and for each metric, we report the mean and standard deviation. The seeds govern the base model initialization and randomization of evaluations; and the SCI controller's responses are deterministic if given a stochastic sampling process.

D. Implementation Details

The SCI controller is the light wrapper of the lead model. During inference, the wrapper calls the model multiple times, with dropout enabled, computing the entropy-based *SP* signal, and checking the stopping condition. The controller maintains an exponentially weighted average of *SP* for threshold adaptation and logs the sequence of *SP* values and step counts for each example.

The value of the threshold γ is equal to α times MAD, with MAD being the median absolute deviation for *SP* over the most recent 50 examples, and where α is a suitably chosen domain specific scaling parameter. We have set the value alpha to be 0.5 for MNIST, 0.3 for MIT-BIH and 0.2 for bearings. The values mentioned are an empirical choice that balances accuracy and compute.

We set the maximum budget $T_{\max}$ to 15 for MNIST and 25 for the other two domains. The values were chosen to allow for sufficient sampling without too much calculation. In practice, the budget is rarely exhausted by the controller, except in the presence of severe distribution shift or ambiguity.

V. EXPERIMENTAL RESULTS

The empirical analysis of the SCI framework in three domains is presented in this chapter. The findings of our work are summarized under the headings of four research questions: (i) metacognitive allocation, (ii) safety signal quality, (iii) compute-accuracy trade-offs and (iv) boundary conditions and failure modes. The results show that SCI spends more computation on hard cases, produces a useful safety signal for informing errors, and achieves efficient compute-accuracy trade-offs.

A. Metacognitive Allocation of Computation

The first inquiry poses a question as to whether SCIs actually "think longer" on hard cases. In Figure 3, we see distribution of inference steps for MNIST and MIT-BIH datasets. it is split based on prediction correctness. Cross-dataset step statistics are summarized in Table I.

In all three domains, SCI consistently expends significantly more computation on examples it gets wrong than on those it gets right. On MNIST, the controller allocates on average 10.31 steps (as opposed to 2.84 steps for correctly classified digits) inferring allocation ratio of about 3.6×. The bearings dataset shows an even higher ratio whereby the faults are predicted using 22.14 steps compared to the healthy windows with 5.83 steps. Thus, the ratio is approximately 3.8×. The ratio is more modest at 1.4× on MIT-BIH, which is perhaps due to the higher baseline difficulty of arrhythmia detection [**?**]

The distributions shown in Figure 3 present an important qualitative pattern where correctly classified inputs tend to stop early, often after a few steps, whereas misclassified inputs are more likely to continue sampling up to the maximum budget. This suggests that the controller is not just doling out extra steps willy-nilly, but that it is sending the extra computation to where it is most needed.

Behavior of the metacognitive allocation is quite interesting on the bearings dataset where the base model achieves almost 100 percent accuracy. Although the error rate is low, the controller allocates much more computation to the few misclassified windows. It indicates that this *SP* signal captures true interpretability difficulty, even when the model is perfectly accurate on average.

B. Safety Signal Quality

The second of the research question asks whether or not, Δ*SP* can become a useful indicator of the failure of the underlying model. For each datasets, we treat misclassification as binary event and compute AUROCΔ*SP*(area under the ROC curve) for predicting a misclassification event where the score is based on Δ*SP*.

On MNIST, we find Δ*SP* has an AUROC of 0.63. The entropy-based signal works well when classifying ambiguous digits, however, for this task, which is quite easy, the classifier assigns most examples a highly confident and correct label. The AUROC on MIT-BIH reaches 0.70, indicating its greater baseline difficulty than the ECG task. On the bearings dataset, the model of ΔSP is able to achieve an AUROC of 0.86 [11].

The risk-coverage assessment on MIT-BIH gives more evidence on the usefulness of the safety signal.

We began with the full test set (error rate 13.22%) and progressively rejected cases which had the largest ΔSP, i.e., in which the controller struggled to reach its target. With 52.6% coverage retained, the error rate falls to 6.24%, which indicates a 52.8% relative decline in risk. This means that in safety-critical situations, ΔSP can either restrict predictions or cause escalation of predictions, reducing number of errors made by predictions while withholding some of the predictions [15].

C. Compute-Accuracy Trade-offs

The third research question asks whether SCI can achieve better compute-accuracy trade-offs than a fixed-budget ensemble. We compare SCI to a baseline that runs the model for a fixed number of stochastic passes and averages the outputs, with no feedback from $SP(t)$. A sweep over fixed budgets $K \in \{1, 2, 4, 8, 16\}$ yields accuracies between 90.2% and 91.6% on the ECG task, with costs exactly equal to K.

The accuracy achieved by SCI is 92.1% in an average of 14.6 steps which slightly outperforms the fixed ensemble with K=16 which has an accuracy of 91.5% and requires 16 steps. These numbers may not seem impressive but they show that our controller can do as good or better than a strong fixed-budget ensemble but with less average computation. The controller stops early on easy cases and spends more time on hard ones.

Gains are more efficient when it comes to reducing variance. Compared to a fixed-budget ensemble of similar cost, the predictions of SCI are more stable because the controller ensures that the interpretive state has become sufficiently clear before the stopping. When consistency is essential, then the recipe of a safety-critical application varies drastically [7].

D. Boundary Conditions and Failure Modes

The fourth research concern focuses on understanding the SCI mechanism breakdown. We experiment with heavy Gaussian noise added to the ECG input to represent extreme out-of-distribution (OOD) conditions. Under severe corruption, the classifier outputs become nearly constant, while the controller terminates after only a few additional steps (7-8). Consequently, ΔSP achieves an OOD AUROC of approximately random performance (~ 0.46). In this regime, SCI correctly learns to "give up" early in terms of computation; however, the resulting signal is no longer discriminative.

This behavior is the opposite of the ambiguous regime described above: when the signals are weak but structured, ΔSP grows and the controller spends more steps; then, when the signal is totally destroyed, the base model and SCI saturate. It appears that SCI is the most informative as a regulator for complex but intelligible inputs than for a universal OOD detector [8].

The policy based on threshold of the controller is sometimes too conservative, stopping too early for inputs that would need longer to resolve. This is particularly noticeable on the MIT-BIH dataset, where some arrhythmic patterns bear visual similarity to normal patterns and therefore require many samples for distinction. The parameter $T_{\max}$ of the optimal budget limits the controller from going (spending) too many steps, but the accuracy degrades in cases where $T_{\max}$ is not sufficient.

VI. DISCUSSION AND CONCLUSION

According to the results shown in Section V, the Surgical Cognitive Interpreter (SCI) framework can enable existing deep learning models to possess a basic level of metacognitive regulation, which is to signal ambiguity, allocate computational resources appropriately, and indicate uncertainty in a manner that can be utilized by downstream systems. In each of the three domains, SCI assigned more inference steps to misclassified inputs which produced a useful safety signal for error detection and a useful compute-accuracy trade-off.

The allocation behavior metacognitive is perhaps the most shocking result. The controller using 3.6× to 3.8× more steps on errors than on correct predictions strongly suggests that the entropy-based SP signal does reflect difficulty in interpretation. The controller doesn't just stop the sampling when it has high confidence, rather this is a dynamic sampling process that adapts to the evolution of the interpretative state. It is not about the confidence of the underlying base model.

The quality of the safety signal may vary over domain; nevertheless it suffices in practice. The AUROC values of 0.70 on MIT-BIH and 0.86 on bearings imply that ΔSP contains important information about the failure probability. The ECG task's risk-coverage sweep shows a concrete use case: by rejecting high-ΔSP cases, it halves error rate, which is significant in safety-critical settings where false positives can be costly.

The compute-accuracy trade-offs imply that SCI is not just using more compute to improve accuracy, but rather using compute better. Through a considered allocation of calculations dependent on interpretive difficulty, SCI produces slightly better accuracy than an ensemble of similar cost that has a fixed budget and the bonus of stable, consistent predictions [1].

The current work has several limitations that need acknowledgment. To begin with, we concentrated on an uncomplicated implementation of the SCI, using entropy as an interpretive signal. More comprehensive

ideas of *SP*, including domain consistency, semantic grounding, and predictive alignment, may be stronger. Moreover, the evaluation has been limited to classification tasks; it remains open whether SCI can be extended to regression tasks structured outputs high-capacity architectures. Moreover, the human-in-the-loop extension has not been tested empirically in the present work; future work should assess the influence of operator feedback on system behavior [2].

TABLE II
COMPARISON OF SCI TO FIXED-BUDGET ENSEMBLE ON THE MIT-BIH ECG TASK

Method	Accuracy (%)	Avg. Steps
Fixed ensemble ($K = 16$)	91.5	16.0
SCI ($SP^* = 0.85$)	92.1	14.6

SCI achieves slightly higher accuracy with lower average compute by allocating steps adaptively.

In several directions, future work can extend the SCI framework. We will first explore more sophisticated definitions of *SP*, whose full SCI formulation includes domain consistency and predictive alignment, among others. Next, we will broaden the framework to the larger architecture and the more challenging domains such as language modeling and medical imaging [3]. Next, we examine SCI as a bridge to causal inference. The interpretive state must capture not only association, but also directed causal relation. In Stage 4, we will carry out prospective deployment studies in clinical and industrial settings to assess SCI's impact on operator trust, decision quality, and safety outcomes [4].

In conclusion, this study offers a lightweight and model-agnostic path towards the safe application of deep learning. The SCI framework shows that a simple controller relying on an appropriately chosen scalar entropy measure can imbue a model with metacognition-like capabilities. Simple cases are resolved at a low cost, while hard cases lead to prolonged decision-making. In addition to this, the mechanism produces a clear expression of uncertainty, which can be acted upon by downstream systems. Closed-loop regulation of inference is necessary when delegating to AI in safety-critical environments, we believe [6].

REFERENCES

[1] P. Muppuri, "Orchestrating Distributed Deep Learning atop Spark: A Systems-Centric Pathway for Scalable Big Data Training," 2025. doi: 10.1109/MEDCOM67532.2025.11405074.

[2] S. Korde, "AI-Driven Exercise Monitoring for Non-Surgical Obesity Treatment: A Deep Learning-Based Posture and Movement Optimization System," 2025. Available: https://journals.femington.com/ijcrms/article/view/126.

[3] S. Tendulkar, "Enhancing Differential Privacy in Deep Learning: Efficient Techniques for Training Large Models," 2025. Available: https://ieeexplore.ieee.org/document/11005164.

[4] Y. S. Kamma, "Intelligent Vision-Based Framework for Dynamic Urban Flow Prediction Using Deep Learning Architectures," 2025. doi: 10.21203/rs.3.rs-8662362/v1.

[5] F. M. Patel, "Fire Protection Strategies in Mechanical Systems: Design, Implementation, and Impact on Safety," 2025. Available: https://www.ijisrt.com/special/fire-protection-strategies-in-mechanical-systems-design-implementation-and-impact-on-safety.

[6] P. Panwar, "DeepSched: Neural Network-Based Task Scheduling for Heterogeneous Distributed Systems," 2025. Available: https://ieeexplore.ieee.org/abstract/document/11339673.

[7] V. Chitukoori, "Interpretable Neural Networks: Bridging Generalized Additive Models and Deep Learning for Enhanced Transparency and Flexibility," 2025. Available: https://www.computer.org/csdl/proceedings-article/sera/2025/11154505/2a3zaIeuEJa.

[8] T. Gupta, "High-Performance Deep Learning for Large-Scale Image Recognition: A GPU-Accelerated Convolutional Approach," 2025. Available: https://ieeexplore.ieee.org/document/11409295.

[9] A. Sharma, "Cognitive Computing Frameworks for Financial Decision Systems: A Multi-Paradigm Synthesis," 2025. Available: https://journals.femington.com/ijambi/article/view/231/42.

[10] A. Patel, "Enhancing Intrusion Detection Systems through Advanced Feature Engineering and Machine Learning Techniques," 2025. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=\&arnumber= 10912389\&isnumber=10912086.

[11] B. N., "Interpretable Deep Learning for Biological Age Prediction: A Counterfactual Approach to Personalized Health Insights," 2025. Available: https://www.atlantis-press.com/proceedings/icsiaiml-25/126021226.

[12] S. Shivankar, "Deep Learning-Based Calibration Pattern Detection for Enhanced RGB-LiDAR Sensor Fusion," 2025. doi: 10.36227/techrxiv.175695799.96992725/v1.

[13] K. S. Penumarthi, "Computational Prediction of Directional Drug-Drug Interactions Using Similarity Fusion and Deep Learning," 2025. Available: https://ieeexplore.ieee.org/abstract/document/11232139.

[14] S. K. Bommakanti, "Artificial Intelligence: Adaptive Certainty Propagation in Distributed Cognitive Systems," 2025. doi: 10.67231/r4n6hk98.

[15] G. Sankaran, "A Hybrid Deep Learning Architecture with Retrieval-Augmented Explanation Generation for Interpretable EEG-Based Stress Classification Across Multiple Paradigms," 2025. doi: 10.67231/smjcrk47.

[16] J. Gupta, "Exploring Deep Reinforcement Learning Algorithms: From Theory to Practice," 2025. Available: https://link.springer.com/chapter/10.1007/978-981-96-9494-5_12.

[17] S. Tendulkar, "Acute Lymphoblastic Leukemia Blood Cell Classification Using Federated Learning," 2025. Available: https://ieeexplore.ieee.org/document/11436284.

[18] S. Daryanavard, "Real-time predictive artificial intelligence: Deep reinforcement learning for closed-loop control systems and open-loop signal processing," Ph.D. dissertation, University of Glasgow, 2024.

[19] N. Kadom, T. S. Cook, and M. A. Bruno, "Improving diagnosis: Advances in radiology," *Diagnosis*, vol. 12, no. 4, pp. 578–587, 2025.

[20] R. Bhidayasiri and W. Maetzler, Eds., *Handbook of Digital Technologies in Movement Disorders*. Elsevier, 2024.

[21] K. Amunts, M. Axer, S. Banerjee, L. Bitsch, J. G. Bjaalie, P. Brauner, et al., "Royal Netherlands Academy of Arts and Sciences (KNAW)," *Imaging*, vol. 2, 2024.

[22] V. Ambila, "Exploring Quality Management Strategies to Mitigate Culture Shock Among Cochlear Implant Users," Ph.D. dissertation, Walden University, 2021.