# Resolute Resolutions (a blog), volume 4:
# AI ≔ Introspection+Updatability+Uncertainty

**July 2022**

**Allen Schmaltz**

## Abstract

We highlight three themes that encapsulate the key properties necessary for engendering trust in neural networks, and engendering trust in the ability of end-users to interact with neural networks: INTROSPECTION, UPDATABILITY, and UNCERTAINTY.

At a coarse resolution, we view the goal of AI research as building tools to assist humans in discovering new scientific knowledge, accelerating the removal of the masks temporarily put before our eyes by nature. We seek to build a *learned database*[1] to which we can pose a query that is then returned to us as a relative distance comparison to all existing knowledge. Unlike a traditional database in which we seek provenance via an exact match, in this case we seek provenance via interlocking, consistent distance relations across examples. In contrast, single point predictions from an all-knowing blackbox are not of interest to us; instead, we seek to extend our own knowledge of the world via the assistance of reasoning through learned analogies against the existing knowledge base. This abstraction lends itself to a path toward building AI systems that are interpretable and reliable. It provides a way of conceptualizing how we might avoid errors in highly-interconnected systems that could otherwise unexpectedly scale out of control.

More specifically, with this abstraction as our guidepost, we briefly highlight three themes that encapsulate the key properties necessary for engendering trust in the current generation of neural networks, and engendering trust in the ability of end-users to interact with neural networks: INTROSPECTION, UPDATABILITY, and UNCERTAINTY.

A contemporary deep network, $F$, regularly consists of billions (soon, trillions) of real-valued parameters.[2] Such models are of particular interest for tasks with high-dimensional inputs, such as text, amino acids, images, and/or videos, over which we aim to predict a label on some new, unseen input. To assign values to those parameters via supervised learning, we have access to a training dataset, $\mathcal{D}_{\text{tr}} = \{(X_i, Y_i)\}_{i=1}^{I}$ of $|\mathcal{D}_{\text{tr}}| = I$ instances paired with their corresponding ground-truth labels, $Y_i$. For the purposes here, we will assume that the outcome is discrete,[3] $Y_i \in \mathcal{Y} = \{1, \ldots, C\}$, $C \in \mathbb{N}^+$. We are similarly given a held-out labeled calibration dataset, $\mathcal{D}_{\text{ca}} = \{(X_j, Y_j)\}_{j=I+1}^{N=I+J}$ of $|\mathcal{D}_{\text{ca}}| = J$ instances. We are then given a new test instance, $X_{N+1}$, from an unlabeled test set, $\mathcal{D}_{\text{te}}$, and we seek to predict the unseen label $Y_{N+1}$, as close as possible to the theoretical ideal of Bayes/irreducible error, i.e., up to the underlying uncertainty in the environment. The learning procedure consists of assigning *particular*—but typically not uniquely identifiable, one-to-one—values to the parameters of $F$ via systematic perturbation of the values using samples from $\mathcal{D}_{\text{tr}}$ and comparing the true $Y_i$ against the predicted $\hat{Y}_i$, typically via empirical

[1] As a point of contrast, it may be productive to call this an Artificial General Comparatist (AGC), as opposed to an Artificial General Intelligence (AGI). In any case, from where we now stand, the distance is very far to these ends, but the key takeaway is that by decoupling from the goal of emulating humans, we get a practical roadmap for managing and deploying the *current* generation of networks. In fact, in the more mundane world of the existing deep networks, we have *already shown* how to achieve key elements of the three themes described in this post.

[2] More specifically, we have in mind that $F$ is a large Transformer network pre-trained over very large amounts of data, and possibly further fine-tuned with labeled data for a particular task.

[3] In principle, we can treat regression as classification over discretized bins, after which the approaches we have proposed in our recent works may be directly applied.

risk minimization. In this way, we arrive at a—potentially high-accuracy—blackbox, to which we can feed new inputs.

Provided we can observe high-accuracy predictions on our held-out set using $F$, it may seem our job is done, even if the parameters are not identifiable. That is to say, we might repeat the learning procedure and arrive at different values of the parameters of $F$, potentially even with identical training error on $\mathcal{D}_{\text{tr}}$, but we may nonetheless be content with sufficiently high accuracy on our held-out set $\mathcal{D}_{\text{ca}}$. In fact, relying on the point predictions alone of such a model is far from sufficient for higher-risk settings.

First, the datasets $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{ca}}$ we saw when initially learning may turn out to be rather different than those of the new, previously unseen $\mathcal{D}_{\text{te}}$ encountered at test. We can think of each of these datasets being themselves samples (possibly biased) from $\mathbb{D}$, the unknown underlying data distribution of the environment. Distributions unlike those seen at training may throw the whole endeavor off the rails. Thus, we must be able to estimate the degree to which inputs differ from those seen in training, comparing distributions in high-dimensions, which is challenging, in general. Additionally, even for data similar to that which we have seen in the past, we need to be able to produce reliable estimates of uncertainty over the predictions, given the context of the particular environment.

Unfortunately, these challenges preclude a randomized-trial-style approach using point predictions. The data is simply too high-dimensional and the non-identifiable parameters too numerous to rely on such an outlook alone. For example, in tests, we may find that our network performs well on the observed population samples, but unless we have some facility for interpreting the predictions and their uncertainty, we may not be able to know when subtle—or even not so subtle—shifts in the input distribution occur, potentially rendering the model useless, or in high-risk settings, unwittingly dangerous.

How then to approach this blackbox of millions, billions, or trillions of non-identifiable parameters? In a series of works (see, e.g., Schmaltz and Rasooly, 2022, the most recent entry), we have proposed, and provided approaches for, characteristics that can be distilled into the following three themes:

1. INTROSPECTION: With our learned set of non-identifiable parameters of $F$, we want to be able to cut the graph of parameter applications (against the input) relative to the features of interest at the desired unit-of-analysis (determined by the end-user), mapping those at inference to those with known labels from $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{ca}}$ (hereafter, the "support" set). Central to this is modeling the relationship of those features to the predictions at the unit-of-analysis of the task (and vice-versa).

2. UPDATABILITY: We want to be able to allow local, end-user updates to the mappings established in (1) without altering the original set of non-identifiable parameters (i.e., we seek local "patches" without creating a cascade of unintended and unexpected global side-effects, such as catastrophic forgetting).

3. UNCERTAINTY: We seek reliable uncertainty estimates at the unit-of-analysis of available labels in the support set.

To achieve (1)-(3), we need to determine the inductive (architecture) biases amenable to decomposing the network into human-understandable parts for a given task, as with class-conditional feature detection via hard attention mechanisms, for example. This step is task-specific; for example, in some settings, we may only be concerned with modeling at the instance (global) level, whereas in other cases we might seek a sparse set of features that determine, contribute to, or otherwise serve as summarizations of, the global prediction. We then approximate the parametric model, at the desired unit-of-analysis, as an instance-based metric learner, which becomes the model we use in practice, and from which we derive a mapping to examples with known labels and uncertainty estimates.

We want, in some sense, for (1)-(3) to still be valid for any dataset sampled from $\mathbb{D}$, even if the sample is not particularly representative. That is to say, we do not want the whole enterprise to fall apart with distribution shifts possible within the environment of a given task. Reliably predicting over all of the unobserved subsets of $\mathbb{D}$ will be hard, in general, but at least we can try to restrict ourselves to input similar to the support set, assigning greater uncertainty as we get farther from the preponderance of the mass of the observed instances in the support set.

Achieving (1)-(3) has important implications for equity in the prediction systems that are determining, and are anticipated to determine, many high-stakes decisions in the years ahead. Because distribution-free conditional coverage[4] is impossible in general (see Schmaltz and Rasooly, 2022), invariably, we will post-hoc analyze our data and find in some cases that some previously unconsidered subset of protected attributes has been undercovered. However, at least we can use (1) to analyze the data relative to the predictions; we can use (2) to update labels and/or add additional examples; and then we can stratify and obtain coverage of the new protected attributes via (3). A reasonable and likely achievable normative goal for application deployments is to strive for similar coverage across known subsets of the population. In contrast, uniform *cardinality* of the prediction sets will be difficult to achieve across known subsets, but at least with larger sets, one knows to send to a human for adjudication, collect more data, and more generally, to analyze the data further using the aforementioned characteristics.

One of the substantive surprises in computer science and statistics in recent years in our opinion is that we can obtain these characteristics (to at least some degree) in some single (but nonetheless, non-trivial and of interest in practical settings) task environments, by leveraging the dense representations of the very large parametric networks for matching, constructing approximations that are at least as effective as the original underlying parametric network. This comes as a surprise in part because the curse of dimensionality might seem to preclude such approaches and behavior. However, it turns out the deep networks have inductive biases conducive to being decomposed into effective metric learners for matching within the observed space. When this behavior is combined with datasets that are very large, but nonetheless now available for some tasks, we can achieve characteristics (1)-(3).

Characteristics (1)-(3) serve as an interlocking set of checks and balances on the model, and provide an actionable understanding of the predictions. Interestingly, in addition to providing desirable properties for end-users not necessarily otherwise familiar with neural networks, this behavior also points to a rather different trajectory for how scientists and engineers approach both analysis and discovery.[5] It seems for some tasks moving forward there will be an inflection point at which we have sufficient *trust* in achieving (1)-(3) by the model and *trust* in end-users' abilities to interact with (1)-(3) for the deep networks that we prefer them over simpler models (e.g., linear models or Bayesian hierarchical models). Prospectively there is a point at which you have enough data and sufficient achievement of (1)-(3) that certain high-risk settings are more reliable in practice to just learn directly via architecture choice and then approximate as instance-based metric learners, enabling (1)-(3), than to try to model at the outset with simpler functions with identifiable parameters, since the latter nonetheless require and are sensitive to non-trivial user tuning parameters, engineering choices, and prior beliefs. In these cases, the deep networks then become the more trustworthy modeling choice. More generally, there are many open problems in the data-oriented sciences in which we have large amounts of coarsely-labeled data points but limited physical or statistical models of their behavior; leveraging characteristics (1)-(3) with the large deep networks may be particularly productive for these problems.

## References

Allen Schmaltz and Danielle Rasooly. 2022. Approximate Conditional Coverage via Neural Model Approximations. *arXiv:2205.14310*.

[4] Rather than predicting a single $\hat{Y}_{N+1} \in \mathcal{Y}$, we seek to construct a prediction set, produced by a set-valued function $\hat{\mathcal{C}}(X_{N+1}) \in 2^C$, containing the true unseen label with a specified coverage level $1 - \alpha \in (0, 1)$ on average.

[5] Broadly speaking, we separate real-world AI classification problems into two types. In Type I ("consumer tasks"), we are only interested in predicting over safely in-distribution data. These are typical consumer applications; in these cases, we simply refrain from predicting over unusual inputs, which we send to humans for further adjudication. In Type II ("discovery tasks"), more commonly encountered in the natural sciences (e.g., various tasks related to protein folding), we ignore *both* closely in-distribution data *and* far out-of-distribution data, aiming to minimize the set of remaining instances between those two extremes to send to humans for further investigation—in the hope of more efficiently uncovering new scientific knowledge than traditional methods.