Perspectives on incorporating expert feedback into model updates

Valerie Chen,1,* Umang Bhatt,2,3,* Hoda Heidari,1 Adrian Weller,2,3 and Ameet Talwalkar1
1Carnegie Mellon University, Pittsburgh, PA, USA
2University of Cambridge, Cambridge, UK
3The Alan Turing Institute, London, UK
*Correspondence: valeriechen@cmu.edu (V.C.), usb20@cam.ac.uk (U.B.)
https://doi.org/10.1016/j.patter.2023.100780

SUMMARY

Machine learning (ML) practitioners are increasingly tasked with developing models that are aligned with non-technical experts’ values and goals. However, there has been insufficient consideration of how practitioners should translate domain expertise into ML updates. In this review, we consider how to capture interactions between practitioners and experts systematically. We devise a taxonomy to match expert feedback types with practitioner updates. A practitioner may receive feedback from an expert at the observation or domain level and then convert this feedback into updates to the dataset, loss function, or parameter space. We review existing work from ML and human-computer interaction to describe this feedback-update taxonomy and highlight the insufficient consideration given to incorporating feedback from non-technical experts. We end with a set of open questions that naturally arise from our proposed taxonomy and subsequent survey.

INTRODUCTION

Before deploying a machine learning (ML) model in high-stakes use cases, practitioners, who are responsible for developing and maintaining models, may solicit and incorporate feedback from experts.1–3 Prior work has largely focused on incorporating the feedback of technical experts (herein, ML engineers, data scientists, etc.) into models.4–10 The feedback of technical experts might be immediately actionable, as likely few communication barriers exist between technical experts and practitioners. In contrast, the relationship between a practitioner and a non-technical expert (herein, doctors, lawyers, elected officials, policymakers, social workers, etc.), as illustrated in Figure 1, is more complex.11,12 Upon seeing information about the model, the expert provides feedback based on their preference to practitioners, who can then update the model. There has been insufficient consideration of how to incorporate feedback from non-technical, domain experts13,14 into models. To bridge this gap, we start by examining model updates available to the practitioner and the types of feedback that non-technical experts might provide. We clarify the mechanisms available to turn feedback into updates and then devise a taxonomy along two axes: (1) levels of expert feedback and (2) types of model updates. Along the first axis (levels of domain expert feedback), expert feedback may come as domain-level feedback, which captures high-level conceptual feedback that the practitioner must translate into updates, or observation-level feedback, which captures how the model should behave on a few, specific data points.15–18 Along the other axis (types of model updates), we consider the updates a practitioner can make to a supervised

THE BIGGER PICTURE

Machine learning practitioners develop and deploy models that affect millions of people worldwide. Data scientists often receive feedback from domain experts like doctors, account executives, or regulators, all of whom may opine on a model and its outcomes. We ask: how can a data scientist be empowered to fix models to align better with the goals and values of experts? We provide an extensive review of methods for updating models based on expert feedback. For example, a radiologist could give feedback by drawing bounding boxes of where, in each X-ray, a specific fracture lies or by providing additional electronic health record data to co-reference a given scan. This feedback can then be used to augment the dataset to make a model more robust, alter the training objective to account for compliance objectives, or edit parameters to ensure that the model relies on the correct features. This review closes with concrete open questions to better involve non-technical experts in model development that are ripe for future work.

OPEN ACCESS

Patterns 4, July 14, 2023 © 2023 The Author(s). This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
learning objective, where feedback typically changes the dataset, the loss function, or the parameter space. (Supervised learning covers a broad range of model classes, ranging from vision transformers, large language models, and impactful application areas, like medical diagnostics and criminal justice. We consider other objectives, which may include reinforcement learning or unsupervised learning, to be out of scope for this review. We focus on methods more commonly deployed in practice and omit the Bayesian analog for our pipeline, where experts can express preferences over the distribution of functions.) The two axes of our taxonomy form six distinct categories for feedback-update interactions. We place existing work from human-computer interaction, ML, robotics, and FATE (Fairness, Accountability, Transparency, Ethics) into each of these categories in mapping prior work to our taxonomy.

Our taxonomy not only provides a preliminary understanding on the ways in which non-technical expert feedback can be converted into practical model updates but also motivates a diverse set of open questions to improve practitioner-expert interactions. In open questions, we pose questions on how to connect model information to our taxonomy, how to prompt and elicit expert feedback effectively, and how to decide on the type of update to perform given the feedback. We hope that our taxonomy grounds the community in concrete ways to leverage non-technical expert feedback in a practical way while still encouraging future research to further feedback incorporation.

**FEEDBACK-UPDATE TAXONOMY**

One role of practitioners is to convert non-technical expert feedback into a model update. (We note the important case that, sometimes, valuable expert feedback might be received to say that using any model is not appropriate for the setting at hand. While such a concern must be taken seriously and considered carefully with relevant stakeholders, we do not discuss this case further here.) We describe the diverse ways that expert feedback can lead to model updates through our feedback-update taxonomy (Table 1). While experts are often involved prior to training an initial model, we focus on the iterative feedback process after a model has been trained. One piece of feedback could be used to alter multiple parts of the objective (e.g., change the dataset and loss function), but each update should be considered individually. We first elaborate on the two axes of our taxonomy and flesh out each category in the next section.

**Levels of domain expert feedback**

Once an expert has observed information about the model, practitioners may ask for feedback to improve the model’s behavior in two general ways.

- Domain feedback: it may be natural for non-technical experts to provide high-level conceptual feedback. The expert could provide explicit feedback over a set of good models or suggest data pre-processing to reduce discrimination.
- Observation feedback: it may also be possible to learn by observing expert behavior. For example, practitioners can use observations to approximate a property of interest (e.g., fairness) or can collect contextual information, where every data point is accompanied by auxiliary information that can be used during learning (e.g., feature attributions, style factors, and semantically meaningful concepts).

These two types of feedback form one axis of our feedback-update taxonomy in Table 1. While these two forms of feedback may be non-exhaustive, they capture a wide variety of mechanisms for non-technical experts to influence the development of models. Neither type requires non-technical experts to have knowledge about the model or the training process itself. A radiologist could provide domain-level feedback about X-ray scans via high-level information about the region of interest in each X-ray or pre-processing suggestions for every scan. Examples of observation-level feedback that the same radiologist could provide on X-ray scans include bounded boxes of where in each X-ray a specific fracture lies or additional electronic health record data to co-reference a given scan. The role of
practitioners may be expanding, as a practitioner may need to decide, as part of the model update process, whether to treat the collected feedback as domain- or observation-level feedback.

We consider other forms of feedback to be out of scope for this work because they are less intuitive to elicit from a non-technical domain expert.27,82 This includes changing the learning algorithm (e.g., in differential privacy communities9,84), selecting hyperparameters (e.g., in AutoML research5,85), and specifying the order of data points given to a learning algorithm (e.g., in machine teaching literature6,8,10).

Types of model updates
In the supervised learning setting, a practitioner generally minimizes a loss function on a provided dataset to learn the parameters of a model. Once experts have provided feedback, practitioners can leverage expert input to improve the model in multiple ways: updating the dataset, the loss function, or the parameter space. These update types form the other axis of Table 1.

- Dataset updates: feedback can be incorporated by changing the dataset on which the model is trained.
- Loss function updates: feedback can also be incorporated by adding a constraint to the optimization objective. This manifests as a change in the loss function.
- Parameter space updates: finally, feedback can also be provided on the parameters of the model itself, which reflect a change in the parameter space.

For example, a public official may ask that as the input feature population increases, the likelihood of a project proposal getting funded should increase; this implies monotonicity between an outcome and input feature. Practitioners can incorporate this feedback in various ways, which are different technically. The practitioner can update the dataset by adding or removing appropriate data points, update the loss function by adding a regularizer that penalizes the model for not satisfying this condition, or update the parameter space by optimizing over a subspace of parameters that satisfy this condition.

While these update types may seem straightforward, it is unclear how to identify the new dataset, loss function, or model parameter space: this involves transforming domain expert feedback into one of these three general updates. After the practitioner incorporates expert feedback, the updated model should ideally reflect the expert’s preferences better than the original model. In the next section, we flesh out the conversion from feedback to update.

MAPPING PRIOR WORK TO OUR TAXONOMY

For each feedback-update pair, we illustrate how the practitioner can use the feedback to modify the ML model. We provide key takeaways for each feedback-update pair, summarizing current work and future directions. To identify relevant work in each category of the taxonomy, we used a snowball sampling methodology86 to gather references that pertain to expert feedback incorporation. For clarity, we provide examples for each category to ground our discussion. While each category is its own research direction, we provide a broad, non-exhaustive discussion of how to update models under expert feedback. Note that many references we provide do not consider a domain expert explicitly but can be applied to the feedback that the domain expert provides. While all feedback may not always be immediately actionable, we hope our paper encourages future work to help practitioners address non-technical experts’ needs and concerns more efficiently.

Domain to dataset
A dataset can be modified given domain-level feedback. The practitioner can modify the original data or generate new data. If the expert provides domain-level feedback that the model...
should not rely on a feature, then adding data can lead to a model that satisfies this expert-specified domain-level feedback, as shown in Figure 2.

**Dataset modifications**
A non-technical expert may specify feedback that entails a dataset modification and, consequently, a retraining of the model. One example that naturally suggests a dataset update is dataset pre-processing; here, a new perturbed dataset is obtained by applying a transformation to features of the original dataset. In many cases, the sizes of the new and old datasets are the same. Suppose the expert suggests that the dataset should be balanced in terms of the sensitive attribute; the practitioner can design a transformation function that achieves this property. Extensive work has been done on data pre-processing for fairness. Another example of domain-level feedback that leads to dataset updates is data augmentation, where an augmentation scheme is applied to every dataset point in the original dataset, yielding a dataset of original points and augmented points. In these settings, the size of the dataset tends to be larger after augmentation. After the expert specifies that the model should make the same prediction regardless of image rotation, the practitioner may consider modifying the dataset by adding rotated variants of input images to the dataset, which allow for more robust inference.

**Data generation**
Other domain-level feedback may not necessarily suggest a dataset modification but rather may prescribe a way to generate data to augment the existing dataset. For example, an expert may say that the model should not rely on spurious correlations between people and other objects in images (e.g., using a tennis player to detect the tennis racket). A practitioner may generate counterfactual images from the original dataset that contain and do not contain the people and other objects to decrease reliance on the person in the image. Generative modeling techniques are useful to create synthetic data that adhere to a property specified by an expert. For example, to assuage an expert’s fairness concerns, a practitioner could generate data under a fairness constraint per Xu et al. Weak supervision
Domain-level feedback may be used to handle unlabeled data. Weak supervision applies to any approach that deals with data where only some is labeled. To leverage additional unlabeled data, we can obtain lower-quality labels efficiently. The most common approach is to ask experts to provide higher-level supervision over unlabeled data. For example, an expert may provide an explicit rule stating that “all individuals under 18 should have a negative label.” The practitioner can turn this rule into a pseudo-labeling function, which can be used to leverage large amounts of unlabeled data points.

**Takeaway**
Approaches to modifying or generating data are common in the ML literature. However, these methods do not usually involve eliciting supervision from non-technical experts. Future work should focus on prompting experts for domain-level feedback, which could induce dataset changes in settings where collecting more data may be hard: we elaborate on this in difficulties with feedback collection.

**Domain to loss function**
Practitioners can use an expert’s domain-level feedback to update the loss function. One example is shown in Figure 3.

**Constraint specification**
Experts may have access to properties, beyond performance, that may be well suited to constrained optimization. There has been a diverse body of work on incorporating fairness and interpretability constraints in a usable (e.g., mathematical) format may be challenging for non-technical experts. For example, experts may desire properties that are difficult to convert into a precise statement that can be incorporated in the loss function (e.g., societal norms/ethics).

**Domain to parameter space**
Specified domain-level feedback can reduce the parameter space to a set of potential models. In Figure 4, we show a simple setting where the expert can directly intervene on model weights.

**Model selection**
It may be possible to have the expert provide preferences over a set of models. Though there are a number of open questions...
about how this set is presented (discussed in difficulties with feedback collection), one solution from Lage et al.\textsuperscript{52} is to show an explanation to an expert, calculate the amount of time it takes for the expert to predict the label, and change the model prior (update) accordingly. The change to the model prior manifests as constraints the practitioner places during fine-tuning to ensure that the resulting model is interpretable given a specified tolerance. The model prior can also be implicitly changed by imposing complexity constraints on the set of possible models,\textsuperscript{51,52} which may include constraints on how sparse, interpretable, or smooth the model must be.

**Model editing**

For some model classes (e.g., simple, transparent models\textsuperscript{48,49}), the expert may directly provide feedback that would change the parameters of the model itself without requiring retraining of the model on new data or loss function. For example, Wang et al.\textsuperscript{49} allow experts to directly change the weights of a generalized additive model (GAM) after exposing shape function visualizations and other model properties. In this case, the update to the parameter space is trivial because the set of possible parameters is the exact model that is specified by the expert. These edits on the model might implicitly align the model with unexpressed, desired expert properties, allowing practitioners to avoid sample-inefficient data collection procedures and potential difficulties with constraint specification.

**Takeaway.** While these types of updates traditionally require more technical users, there are increasingly more user-friendly interfaces developed to allow even non-technical experts to edit the model in a more direct manner.

**Observation to dataset**

The dataset can be directly modified from observation-level feedback via actively collecting data (e.g., asking experts to label selected data points) or passively observing behavior (e.g., collecting data from expert behavior in practice). In Figure 5, we show how collecting new data can transform the model.

**Active data collection**

The field of active learning falls into this category.\textsuperscript{92,93} Traditional work on active learning does not explicitly consider the human in the loop (i.e., the choice of new points to add to the dataset are selected by a learning algorithm), and it is rather straightforward to use the updated dataset to retrain the model. Data collection may involve asking the expert to approve the weight placed on a model\textsuperscript{55} or to provide a label for a data point.\textsuperscript{57} Experts may also review new data points, where various experts for each data point.\textsuperscript{51} Recently, active learning has been studied alongside model transparency, specifically using explanations to assist experts with choosing which points to add to the dataset.\textsuperscript{55} Cabrera et al.\textsuperscript{53} propose an extensive visual analytics system that allows experts to verify and produce examples of crowdsourced errors, which can be thought of as additional data.

**Passive observation**

Instead of asking the expert to provide labels on additional pieces of data, the practitioner could also collect data via expert demonstrations. Inverse decision theory argues for observing human decisions to learn their preference.\textsuperscript{18} One can simply observe a non-technical expert’s behavior to generate more data.\textsuperscript{17} For example, a data scientist might choose to wait for radiologists to see more patients before updating the dataset and model.\textsuperscript{19} While this may not be the most efficient way to perform model updates, Laidlaw and Russell\textsuperscript{52} find that forcing humans to make decisions under uncertainty can lead to better preference learning. Note that the online learning community has extensively studied how to incorporate expert knowledge into learning statistical models based on a sequence of observation.\textsuperscript{94–96}

**Takeaway.** Active data collection may not immediately appear to be reasonable feedback to expect from non-technical experts. Compared with the traditional crowd workers used in ML literature, non-technical experts like lawyers and regulators may not have the time to extensively label data that are required for some powerful deep learning methods. However, we do not exclude this work because it may still be desirable to collect data from non-technical experts (e.g., in settings with limited amounts of data) or collect observation-level feedback in a passive manner.

**Observation to loss function**

We identify two ways to collect additional information to learn and integrate a new function into the loss. In Figure 6, we show that practitioners can collect contextual information in the form of error costs to edit the loss function and retrain the model.

**Collecting contextual information**

Experts may have contextual information they wish to share with practitioners. While there is not a unified way for practitioners to incorporate contextual information, it is often used for a loss function update. The contextual information might be provided for some subset of points if collecting it for all data points is expensive.\textsuperscript{97,98} In general, contextual information can be used to constrain the model behavior.\textsuperscript{99,100} Tseng et al.\textsuperscript{73} and Weinberger et al.\textsuperscript{74} both modify the loss function to regularize feature attributions to ensure that the
explanations from models better align with expert expectations. Koh et al.\textsuperscript{70} proposed concept bottleneck models (CBMs) as a way to incorporate pre-defined concepts into a supervised learning procedure, where the concepts are semantically meaningful pieces of information used in a discriminative model to perform prediction.\textsuperscript{101,102} Their supervised approach maps raw inputs to concepts and then maps concepts to outputs. An intermediate layer of a neural network can also be selected as the CBM, where the layer’s activations should be aligned with concepts when training.\textsuperscript{7,71}

Furthermore, expert-specified contextual information can be used in other clever ways. Vapnik et al.\textsuperscript{103} use privileged information for each input to accelerate learning of a support vector machine. Abe et al.\textsuperscript{104} use misclassification costs to find a weighted loss function that improves model performance under class imbalance. Adel et al.\textsuperscript{69} use generative factors to improve deep representation learning. Hind et al.\textsuperscript{72} use explanations to partition classes into subclasses for more accurate models downstream.

**Constraint elicitation**

The practitioner may choose to parameterize a constraint using observation-level feedback. Analogous to metric learning,\textsuperscript{105} this can be done by learning the hyperparameters of a constraint from expert feedback about individual points\textsuperscript{66} or by building a function from expert observations.\textsuperscript{65,106,107} The learned metric is then appended onto the existing loss function. For example, some have attempted to learn an individual fairness constraint after receiving pairwise judgements from experts, who specify if two individuals should be treated the same or not.\textsuperscript{65,107} Practitioners can also constrain intermediate model representations\textsuperscript{67} or edit a model’s representation,\textsuperscript{68} which entails experts selecting exemplar training points that should have similar representations to a test point.

**Takeaway.** The aforementioned approach of constraint specification (see domain to loss function) bears similarity to both approaches we discuss here. The main difference between using constraint specification and collecting contextual information is that contextual information is specified for each data point, which may be easier for an expert to provide generally. Although constraint elicitation techniques circumvent the potential difficulty of constraint specification, these techniques require more creativity in how the individual pieces of observation-level feedback can be combined. We note that there should be further work on organizing these types of approaches, as they likely will be domain specific.

**Observation to parameter space**

In some cases, the parameter space may not be rich enough to find a suitable model that fits the dataset well.\textsuperscript{106} Additional data containing modified features can be used to alter parameter space. In Figure 7, we show how adding a feature to a dataset allows the updated model to better separate the blue and yellow points.

**Feature modification**

This type of update manifests as a change to the parametric form of the model, which now makes predictions on the changed data points. Correia and Lecue\textsuperscript{76} and Roe et al.\textsuperscript{108} use experts to select a subset of features to use for prediction (e.g., to bias the model away from using spurious correlations). Bakker et al.\textsuperscript{75} sequentially add features to a dataset to achieve fairness goals. In these works, the original parameter space may be all two-dimensional models, but the updated parameter space after feature acquisition would be three-dimensional. Moreover, there is a plethora of work on feature selection that implicates the selected model class.\textsuperscript{109,110} There have been works where experts can suggest the model class based on their interpretability needs: the less complex the parameter space, the more interpretable the model.\textsuperscript{111,112}

**Takeaway.** Working with experts to modify features might be particularly helpful earlier in model development. We note that this type of update is already commonly done in practice.

**OPEN QUESTIONS**

Through our review of practitioner-expert interactions, we find there are multiple important and exciting open questions. While our taxonomy highlights many ways to convert feedback to updates, there are still open questions about the entire feedback solicitation and incorporation process, as illustrated in Figure 8. We discuss open questions of how practitioners should prompt and collect feedback from non-technical experts (difficulties with feedback collection) and how practitioners can decide the best type of update to perform given feedback (problems with updating models). We discuss the potential algorithmic and participatory innovations needed to improve our feedback-update pipeline: collaborative effort between multiple communities is imperative to increase non-technical expert involvement in model development.

**Difficulties with feedback collection**

There are many complexities that arise when practitioners collect feedback from non-technical experts, which include prompting for feedback and engaging with multiple experts.
Prompting for feedback

What information about the model should the practitioners present to the expert? What type of prompt should a practitioner choose?

In the process of prompting experts for feedback, practitioners need to decide how and what information about the model needs to be shown to experts. There has been initial work to develop interfaces to visualize trade-offs between multiple objectives, allowing experts to explore model behavior, or to interact with white-box proxies. There has been a flurry of ML monitoring work, which provides dashboards to assess models in production. While these techniques try to bridge the communication gap between practitioners and experts, these tools seldom provide adequate remediation for experts to express their preferences or for practitioners to incorporate those preferences.

Given that experts might want to provide different types of feedback, an important open question is how practitioners should guide non-technical experts via prompts to provide specific feedback. Two such prompt styles include open-ended feedback and forced-choice feedback. In Figure 9, we differentiate how these feedback types can be presented with a simple example.

However, challenges arise both in dealing with the types of prompts and deciding how to word the prompt.

In open-ended feedback, experts are unconstrained in the information they can provide. Thus, the onus is on experts to identify relevant aspects of model behavior on which to opine and to do so in a usable manner. Open-ended feedback is more challenging to use insofar as the feedback may have little to no implication on the model. For example, subjective or qualitative feedback (e.g., measures of confidence or trust) may be difficult to translate into downstream model updates.

Forced-choice feedback is feedback presented to the expert in the form of options, where the expert is forced to pick either one or multiple options or rank the options. There is a related question of how to select, visualize, and present options. For example, in the growing body of work that studies an ε-Rashomon set of good models, there are currently only heuristics to identify this set of models. This may be done by setting multiple random seeds, optimizing under constraints, or subsampling from the dataset; nonetheless, it remains unclear how to visualize this set.

Algorithmic innovation. For open-ended feedback, new methods can let experts better traverse model behavior to improve holistic understanding of the model. For forced-choice feedback, future work can build methods to understand the order in which the queries should be asked and what set of options to provide.

Participatory innovation. The practitioner’s choice of content and presentation of model information will affect the expert’s downstream feedback. Additional user studies are needed to understand the most appropriate pairings of model transparency, prompts, and interfaces to guide the development of useful tools to involve non-technical experts, recognizing potential challenges.

Engaging with multiple experts

How does a practitioner perform an update after receiving feedback from multiple experts?

When non-technical domain experts are considered in our pipeline, there may be multiple, diverse experts who want to express feedback on a model. Thus, it is important to expand our taxonomy to handle multiple domain experts. In an ideal world, there would be an easy way to incorporate feedback from multiple domain experts into models: this is related to participatory ML, a burgeoning community democratizing the ability for all to interact with models.
Practitioners should combine feedback from multiple domain experts carefully, perhaps using an aggregation mechanism. The combinatorial auction literature has long studied how to best elicit preferences when the number of options is combinatorial. Every feedback aggregation mechanism is value laden, as some mechanisms might prefer one expert over others and some tie-breaking mechanisms might under-represent some experts. Even if practitioners had a sensible feedback aggregation mechanism, settling incongruities in feedback is difficult, as there might be conflicting pieces of feedback from multiple, diverse domain experts.

Algorithmic innovation. Developing methods to aggregate feedback from multiple domain experts will require a framework that not only considers the heterogeneous feedback types but also permits various aggregation schemes, which combine potentially contradictory feedback into a collective piece of feedback for the practitioner to incorporate.

Participatory innovation. New methods should consider ways to efficiently elicit feedback at scale from diverse domain experts and mechanisms to make explicit the value-laden aggregation done when ironing out potential contradictions in collected feedback.

Problems with updating models

There is a gamut of considerations when practitioners decide to update models based on expert feedback. We consider when to update based on feedback and which update to use.

When should a practitioner incorporate an expert’s feedback? How can a practitioner best collect feedback to make the most impactful update to the model?

Non-technical domain expert feedback is difficult to collect. As a result, practitioners need to ensure that, given a single piece of feedback, the update they make has a large impact on the model. To measure the impact of an update, there are a few open questions to address. First, it is not clear what impactful means. One naive way is to measure the amount of feedback needed to affect a specified change in the model. However, this ignores how different types of feedback may have different costs of collection and may be amenable to different types of updates.

In settings where feedback suggests that an expert solely cares about performance on a test distribution identical to the training distribution, the practitioner may not be able to make meaningful changes to the model beyond what is learned by the empirical risk minimizer. In some sense, collecting and incorporating expert feedback is only helpful when the expert cares about objectives beyond performance, which may include interpretability, robustness, or fairness. Even after collection, the feedback an expert provided may not always result in a meaningful change to the model. For example, in the late stages of training, additional supervision for in-distribution data or domain-level feedback, which has already been implicitly captured via a sufficiently large amount of labeled data, may prove fruitless.

A practitioner may need to choose between feedback-update pairs because multiple update types could be interchangeable given a piece of feedback. For example, a domain expert’s fairness goals can be achieved by clever sampling from the dataset or by adding constraints to the loss. Others have connected loss function and dataset changes using optimization, Bayesian methods, and group theory.

Algorithmic innovation

There is much work to be done in comparing update types to understand what updates are easier to make for what feedback. Understanding the conditions under which practitioners can use either type interchangeably will be important. From a technical perspective, practitioners can estimate the complexity of an update (e.g., as in Laidlaw and Russell and Zhu et al.).

Participatory innovation

To determine which type of update should be performed, an important factor to consider is the effort required to collect each kind of feedback. Domain-level feedback may require focus groups and workshops. Observation-level feedback might be faster to collect (e.g., large-scale data-labeling platforms), making it easy to collect dataset changes or additions.
Creating efficient ways for practitioners to collect domain-level feedback may reduce the complexity of a loss function update, which might be preferable to dataset updates in resource-constrained settings.

Conclusion
As ML is increasingly deployed in key societal settings, there is a growing need to incorporate domain expert preferences into models. Practitioners need mechanisms to gather and incorporate feedback from non-technical experts into the models they develop. In this work, we studied the interaction between practitioner and expert to see how feedback can be collected and then used to update the model itself. We propose a taxonomy to convert feedback from an expert, who can provide domain- or observation-level feedback, into model updates, which change the dataset, loss function, or parameter space. We conclude with concrete open questions that pertain to prompting for feedback, engaging with diverse feedback, and selecting the update type appropriately. We implore the community to study how to best incorporate domain expertise into the ML development cycle.

ACKNOWLEDGMENTS
The authors would like to thank the following individuals for their advice, contributions, and/or support: Kasun Amarasinghe, Varun Babbar, Alex Cabrera, Katherine Collins, Ken Holstein, Nari Johnson, Emma Kallina, Joon Sik Kim, Jeffrey Li, Vera Liao, Leqi Liu, Bradley Love, Gregory Plumb, Charvi Rastogi, and John Zerilli. U.B. acknowledges support from DeepMind and the Leverhulme Trust via the Leverhulme Center for the Future of Intelligence (CFI) and from the Mozilla Foundation. A.W. acknowledges support from a Turing AI Fellowship under grant EP/V025279/1, The Alan Turing Institute, and the Leverhulme Trust via CFI. A.T. acknowledges support from the National Science Foundation grants IIS1705121, IIS1838017, IIS2046613, and IIS2112471, an Amazon Web Services Award, a Facebook Faculty Research Award, funding from Bocz Allen Hamilton, and a Block Center Grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of these funding agencies.

AUTHOR CONTRIBUTIONS
Conceptualization, V.C. and U.B.; methodology, V.C., U.B., and H.H.; investigation and writing, V.C. and U.B.; review & editing, V.C., U.B., H.H., A.W., and A.T.; project administration, V.C., U.B., A.W., and A.T.

DECLARATION OF INTERESTS
The authors declare no competing interests.

REFERENCES
1. Amershi, S., Cakmak, M., Knox, W.B., and Kulesza, T. (2014). Power to the people: The role of humans in interactive machine learning. AI Mag. 35, 105–120.
2. Cui, Y., Koppol, P., Admoni, H., Niekum, S., Simmons, R., Steinfeld, A., and Fitzgerald, T. (2021). Understanding the relationship between interactions and outcomes in human-in-the-loop machine learning. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 10.
3. Failis, J.A., and Olsen, D.R., Jr. (2003). Interactive machine learning. In Proceedings of the 8th International Conference on Intelligent User Interfaces, pp. 39–45.
4. Adebayo, J., Muely, M., Liccardi, I., and Kim, B. (2020). Debugging tests for model explanations. Adv. Neural Inf. Process. Syst. 33, 700–712.
5. Li, L., Nakandala, S., and Kumar, A. (2021). Intermittent human-in-the-loop model selection using cerebro: a demonstration. Proceedings of the VLDB Endowment 14, 2687–2690.
6. Liu, W., Dai, B., Humayun, A., Tay, C., Yu, C., Smith, L.B., Rehg, J.M., and Song, L. (2017). Iterative machine teaching. In International Conference on Machine Learning (PMLR), pp. 2149–2158.
7. Ross, A.S., Hughes, M.C., and Doshi-Velez, F. (2017). Right for the right reasons: training differentiable models by constraining their explanations. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (AAAI Press), pp. 2662–2670.
8. Simard, P.Y., Amershi, S., Chackering, D.M., Petlon, A.E., Ghoshri, S., Meek, C., Ramos, G., Suh, J., Verhey, J., Wang, M., et al. (2017). Machine teaching: A new paradigm for building machine learning systems. Preprint at arXiv. https://doi.org/10.48550/arXiv.1707.06742.
9. Song, S., Chaudhuri, K., and Sarwate, A.D. (2013). Stochastic gradient descent with differentially private updates. In 2013 IEEE Global Conference on Signal and Information Processing (IEEE), pp. 245–248.
10. Wang, P., Wang, J., Paranamana, P., and Shafto, P. (2020). A mathematical theory of cooperative communication. Adv. Neural Inf. Process. Syst. 33, 17582–17593.
11. Bhatt, U., Xiang, A., Sharma, S., Weiler, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J.M., and Eckersley, P. (2020). Explainable machine learning in deployment. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 648–657.
12. Chen, V., Li, J., Kim, J.S., Plumb, G., and Talwalkar, A. (2022). Interpretable machine learning: Moving from mythos to diagnostics. Queue 19, 28–56. https://doi.org/10.1145/3511299.

13. Bhatt, U., Andrus, M., Weiler, A., and Xiang, A. (2020). Machine Learning Explainability for External Stakeholders (ICML Workshop on Human Interpretability).

14. Kulynych, B., Madras, D., Milli, S., Raji, I.D., Zhou, A., and Zemel, R. (July 2020). Participatory approaches to machine learning. In International Conference on Machine Learning Workshop.

15. Armstrong, S., and Mindermann, S. (2018). Occam’s razor is insufficient to infer the preferences of irrational agents. Adv. Neural Inf. Process. Syst. 31.

16. Davies, K.R. (2005). Inverse Decision Theory with Medical Applications

17. Bhatt, U., Andrus, M., Weller, A., and Xiang, A. (2020). Machine Learning

18. Hertwig, R., and Erev, I. (2009). The description–experience gap in risky

19. Armstrong, S., and Mindermann, S. (2018). Occam’s razor is insufficient

20. Davis, K.R. (2005). Inverse Decision Theory with Medical Applications (Rice University).

21. Hertwig, R., and Erev, I. (2009). The description–experience gap in risky choices. Trends in cognitive sciences 13, 517–523.

22. Swartz, R.J., Cox, D.D., Cantor, S.B., Davies, K., and Follen, M. (2006). Inverse decision theory: characterizing losses for a decision rule with applications in cervical cancer screening. J. Am. Stat. Assoc. 101, 1–8.

23. Irvin, J., Rajpurgar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Bai, R., Shpanskaya, K., et al. (2019). Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In Proc. AAAI Conf. Artif. Intell., 33, pp. 590–597.

24. Piersson, E., Simoiu, C., Overgoor, J., Corbett-Davies, S., Jenson, D., Shoemaker, A., Ramachandran, V., Barghouty, P., Phillips, C., Shroff, R., et al. (2020). A large-scale analysis of racial disparities in police stops across the United States. Nat. Human Behav. 4, 736–745.

25. Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. Adv. Neural Inf. Process. Syst. 30.

26. Coden, A., Danilevsky, M., Gruhl, D., Kato, L., and Nagarajan, M. (2017). A method to accelerate human in the loop clustering. In Proceedings of the 2017 SIAM International Conference on Data Mining (SIAM), pp. 237–245.

27. Guimarães Goecks, V. (2020). Human-in-the-loop Methods for Data-Driven and Reinforcement Learning Systems (PHD thesis).

28. Guo, S., Sanner, S., and Bonilla, E.V. (2010). Gaussian process preference elicitation. Adv. Neural Inf. Process. Syst. 23.

29. O’Hagan, A. (2019). Expert knowledge elicitation: subjective but scientific. Am. Statistician 73, 69–81.

30. Diao, T., Gu, A., Ratner, A., Smith, V., De Sa, C., and Re, C. (2019). A large-scale analysis of racial disparities in police stops across the United States. Nat. Human Behav. 4, 736–745.

31. Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., and Varshney, K.R. (2017). Optimized pre-processing for discrimination prevention. Adv. Neural Inf. Process. Syst. 30.

32. Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. (2015). Certifying and removing disparate impact. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 259–268.

33. Hajian, S., and Domingo-Ferrer, J. (2012). A methodology for direct and indirect discrimination prevention in data mining. IEEE Trans. Knowl. Data Eng. 25, 1445–1459.

34. Iyengar, V.S. (2002). Transforming data to satisfy privacy constraints. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 279–288.

35. Xu, D., Yuan, S., Zhang, L., and Faigin, X.W. (2018). Fairness-aware generative adversarial networks. In 2018 IEEE International Conference on Big Data (Big Data), IEEE, pp. 570–575.

36. Lee, B.C.G., Lo, K., Downey, D., and Weld, D.S. (2020). Explanation-based tuning of opaque machine learners with application to paper recommendation. Free Radical Biology & Medicine.

37. Plumb, G., Ribeiro, M.T., and Talwalkar, A. (2021). Finding and fixing spurious patterns with explanations. Preprint at arXiv. https://doi.org/10.48550/arXiv.2106.02112.

38. Arazo, E., Ortego, D., Albert, P., O’Connor, N.E., and McGuinness, K. (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. In 2020 International Joint Conference on Neural Networks (IJCNN) (IEEE), pp. 1–6.

39. Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J.C., and Liang, P.S. (2019). Unlabeled data improves adversarial robustness. Adv. Neural Inf. Process. Syst. 32.

40. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., and Snorkel, C.R. (2017). Rapid training data creation with weak supervision. In Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, volume 11 (NIH Public Access), p. 269.

41. Ryazanov, I., Nylund, A.T., Basu, D., Hasselöv, I.-M., and Schlip, A. (2021). Deep learning for deep waters: An expert-in-the-loop machine learning framework for marine sciences. J. Mar. Sci. Eng. 9, 168.

42. Coston, A., Rambachan, A., and Chouldechova, A. (2021). Characterizing fairness over the set of good models under selective labels. In International Conference on Machine Learning (PMLR), pp. 2144–2155.

43. Dimanov, B., Bhatt, U., Janmik, M., and Weiler, A. (2020). You shouldn’t trust me: Learning models which conceal unfairness from multiple explanation methods. In Proceedings of the 2020 European Conference on AI.

44. Hiranandani, G., Narasimhan, H., and Koyejo, S. (2020). Fair performance metric elicitation. Adv. Neural Inf. Process. Syst. 33, 11083–11095.

45. Zafar, M.B., Valera, I., Rodríguez, M.G., and Gummadi, K.P. (2017). Fairness constraints: Mechanisms for fair classification. In Artificial Intelligence and Statistics (PMLR), pp. 962–970.

46. Lakkaraju, H., Bach, S.H., and Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1675–1684.

47. Plumb, G., Al-Shedivat, M., Cabrera, Á.A., Perer, A., Xing, E., and Talwalkar, A. (2020). Regularizing black-box models for improved interpretability. Adv. Neural Inf. Process. Syst. 33, 10526–10536.

48. Ustun, B., and Rudin, C. (2019). Learning optimized risk scores. J. Mach. Learn. Res. 20, 1–75.

49. Zhang, H., Morris, Q., Ustun, B., and Ghassemi, M. (2021). Learning optimal predictive checklists. Adv. Neural Inf. Process. Syst. 34, 1215–1229.

50. Frankie, J., and Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In International Conference on Learning Representations.

51. Lin, J., Chen, W.-M., Lin, Y., Gan, C., and Han, S. (2020). Mucnet: Tiny deep learning on iot devices. Adv. Neural Inf. Process. Syst. 33, 11711–11722.

52. Yang, Y., Kandogan, E., Li, Y., Sen, P., and Lasecki, W.S. (2019). A Study on Interaction in Human-In-The-Loop Machine Learning for Text Analytics (IJUI Workshops).

53. Wang, Z.J., Kale, A., Nori, H., Stella, P., Nunnally, M., Chau, D.H., Vorvor-eanu, M., Vaughan, J.W., and Caruana, R. (2021). Gam change: Editing generalized additive models with interactive visualization. Preprint at arXiv. https://doi.org/10.48550/arXiv.2112.03245.

54. Lage, I., Ross, A., Gershman, S.J., Kim, B., and Doshi-Velez, F. (2018). Human-in-the-loop interpretability prior. Adv. Neural Inf. Process. Syst. 31.

55. Dziugaite, G.K., Ben-David, S., and Roy, D.M. (2020). Enforcing interpretability and its statistical impacts: trade-offs between accuracy and interpretability. Preprint at arXiv. https://doi.org/10.48550/arXiv.2010.13764.

56. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. 1, 206–215.
53. Cabrera, Á.A., Druck, A.J., Hong, J.L., and Perer, A. (2021). Discovering and validating at errors with crowdsourced failure reports. Proceedings of the ACM on Human-Computer Interaction 5, 1–22.

54. Fanton, M., Bonaldi, H., Tekiroglu, S.S., and Guerini, M. (2021). Human-in-the-loop for data collection: a multi-target counter narrative dataset to fight online hate speech. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 7 (Long Papers), pp. 3226–3240.

55. Ghai, B., Liao, Q.V., Zhang, Y., Bellamy, R., and Mueller, K. (2020). Learning interpretable concept-bottleneck models. In International Conference on Learning Representations.

56. Wan, S., Hou, Y., Bao, F., Ren, Z., Dong, Y., Dai, Q., and Deng, Y. (2020). Human-in-the-loop low-shot learning. IEEE Transact. Neural Networks Learn. Syst. 32, 3287–3292.

57. Kaushik, D., Hovy, E., and Lipton, Z. (2019). Learning the difference that makes a difference with counterfactually-augmented data. In International Conference on Learning Representations.

58. Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papenot, N. (2021). Machine unlearning. In 2021 IEEE Symposium on Security and Privacy (SP) (IEEE), pp. 141–159.

59. Hiranandani, G., Mathur, J., Koyejo, O., Fard, M.M., and Narasimhan, H. (2021). Optimizing black-box metrics with iterative example weighting. Preprint at arXiv. https://doi.org/10.48550/arXiv.2102.09492.

60. Zhao, S., Fard, M.M., Narasimhan, H., and Gupta, M. (2019). Metric-optimized example weights. In International Conference on Machine Learning (PMLR), pp. 7533–7542.

61. Peterson, J.C., Battleday, R.M., Griffiths, T.L., and Russakovsky, O. (2019). Human uncertainty makes classification more robust. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9617–9626.

62. Laidlaw, C., and Russell, S. (2021). Uncertain decisions facilitate better preference learning. Adv. Neural Inf. Process. Syst. 34.

63. Cheng, H.-F., Stapleton, L., Wang, R., Bullock, P., Chouldechova, A., Wu, Z.S.S., and Zhu, H. (2021). Soliciting stakeholders’ fairness notions in child maltreatment predictive systems. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–17.

64. Hiranandani, G., Boodaghs, S., Mehta, R., and Koyejo, O. (2019). Performance metric elicitation from pairwise classifier comparisons. In The 22nd International Conference on Artificial Intelligence and Statistics, pp. 371–379.

65. Jung, C., Kearns, M., Neel, S., Roth, A., Stapleton, L., and Wu, Z.S. (2019). An algorithmic framework for fairness elicitation. Preprint at arXiv. https://doi.org/10.48550/arXiv.1905.10660.

66. Yaghini, M., Krause, A., and Heidari, H. (2021). A human-in-the-loop framework to construct context-aware mathematical notions of outcome fairness. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pp. 1023–1033.

67. Hilgard, S., Rosenfeld, N., Banaji, M.R., Cao, J., and Parkes, D. (2021). Learning representations by humans, for humans. In International Conference on Machine Learning (PMLR), pp. 4227–4238.

68. Santurkar, S., Tsipras, D., Elango, M., Bau, D., Torralba, A., and Madry, A. (2021). Editing a classifier by rewriting its prediction rules. Adv. Neural Inf. Process. Syst. 34.

69. Adel, T., Ghahramani, Z., and Weller, A. (2018). Discovering interpretable representations for both deep generative and discriminative models. In International Conference on Machine Learning, pp. 50–59.

70. Koh, P.W., Nguyen, T., Tang, Y.S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. (2020). Concept bottleneck models. In International Conference on Machine Learning (PMLR), pp. 5338–5348.

71. Lage, I., and Doshi-Velez, F. (2020). Learning interpretable concept-based models with human feedback. Preprint at arXiv. https://doi.org/10.48550/arXiv.2012.02898.
134. Freeman, R., Brill, M., and Conitzer, V. (2015). General tiebreaking schemes for computational social choice. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pp. 1401–1409.

135. Levin, J., and Nalebuff, B. (1995). An introduction to vote-counting schemes. J. Econ. Perspect. 9, 3–26.

136. Ward, Z.B. (2021). On value-laden science. Stud. Hist. Philos. Sci. 85, 54–62.

137. Brandt, F., Conitzer, V., and Endriss, U. (2012). Computational social choice. Multiagent systems 2, 213–284.

138. Halfaker, A., and Geiger, R.S. (2020). Ores: Lowering barriers with participatory machine learning in wikipedia. Proceedings of the ACM on Human-Computer Interaction 4, 1–37.

139. De-Arteaga, M., Dubrawski, A., and Chouldechova, A. (2021). Leveraging expert consistency to improve algorithmic decision support. Preprint at arXiv. https://doi.org/10.48550/arXiv.2101.09648.

140. Gordon, M.L., Lam, M.S., Park, J.S., Patel, K., Hancock, J.T., Hashimoto, T., and Bernstein, M.S. (2022). Jury learning: integrating dissenting voices into machine learning models. Preprint at arXiv. https://doi.org/10.48550/arXiv.2202.02950.

141. Weinberger, E., Janizek, J., and Lee, S.-I. (2020). Learning deep attribution priors based on prior knowledge. In Advances in Neural Information Processing Systems, volume 33, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, eds. (Curran Associates, Inc.), pp. 14034–14045.

142. Dong, Q., Gong, S., and Zhu, X. (2018). Imbalanced deep learning by minority class incremental rectification. IEEE Trans. Pattern Anal. Mach. Intell. 41, 1367–1381.

143. Suggala, A., Prasad, A., and Ravikumar, P.K. (2018). Connecting optimization and regularization paths. Adv. Neural Inf. Process. Syst. 31.

144. Khan, M.E.E., and Swaroop, S. (2021). Knowledge-adaptation priors. Adv. Neural Inf. Process. Syst. 34.

145. Chen, S., Dobriban, E., and Lee, J.H. (2020). A group-theoretic framework for data augmentation. J. Mach. Learn. Res. 21, 1–71.

146. Zhu, B., Jiao, J., and Jordan, M. (2023). Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models.

147. Dean, S., Rich, S., and Recht, B. (2020). Recommendations and user agency: the reachability of collaboratively-filtered information. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 436–445.