

Designing a Direct Feedback Loop between Humans and Convolutional Neural Networks through Local Explanations

TONG STEVEN SUN, George Mason University, USA

YUYANG GAO, Emory University, USA

SHUBHAM KHALADKAR, George Mason University, USA

SIJIA LIU, Michigan State University, USA

LIANG ZHAO, Emory University, USA

YOUNG-HO KIM, NAVER AI Lab, Republic of Korea

SUNGSOO RAY HONG, George Mason University, USA

The local explanation provides heatmaps on images to explain how Convolutional Neural Networks (CNNs) derive their output. Due to its visual straightforwardness, the method has been one of the most popular explainable AI (XAI) methods for diagnosing CNNs. Through our formative study (S1), however, we captured ML engineers' ambivalent perspective about the local explanation as a valuable and indispensable envision in building CNNs versus the process that exhausts them due to the heuristic nature of detecting vulnerability. Moreover, steering the CNNs based on the vulnerability learned from the diagnosis seemed highly challenging. To mitigate the gap, we designed DeepFuse, the first interactive design that realizes the direct feedback loop between a user and CNNs in diagnosing and revising CNN's vulnerability using local explanations. DeepFuse helps CNN engineers to systematically search "unreasonable" local explanations and annotate the new boundaries for those identified as unreasonable in a labor-efficient manner. Next, it steers the model based on the given annotation such that the model doesn't introduce similar mistakes. We conducted a two-day study (S2) with 12 experienced CNN engineers. Using DeepFuse, participants made a more accurate and "reasonable" model than the current state-of-the-art. Also, participants found the way DeepFuse guides case-based reasoning can practically improve their current practice. We provide implications for design that explain how future HCI-driven design can move our practice forward to make XAI-driven insights more actionable.

CCS Concepts: • Computing methodologies → Learning settings; Machine learning; • Human-centered computing → Human computer interaction (HCI); Interaction paradigms.

Additional Key Words and Phrases: Convolutional Neural Network, Alignable AI, Steerable AI, DeepFuse, IAA

ACM Reference Format:

Tong Steven Sun, Yuyang Gao, Shubham Khaladkar, Sijia Liu, Liang Zhao, Young-Ho Kim, and Sungsoo Ray Hong. 2023. Designing a Direct Feedback Loop between Humans and Convolutional Neural Networks through Local Explanations. 1, 1 (July 2023), 32 pages. <https://doi.org/XXXXXXX.XXXXXXX>

Authors' addresses: Tong Steven Sun, George Mason University, USA, tsun8@gmu.edu; Yuyang Gao, Emory University, USA, yuyang.gao@emory.edu; Shubham Khaladkar, George Mason University, USA, skhaladk@gmu.edu; Sijia Liu, Michigan State University, USA, liusiji5@msu.edu; Liang Zhao, Emory University, USA, liang.zhao@emory.edu; Young-Ho Kim, NAVER AI Lab, Republic of Korea, yghokim@younghokim.net; Sungsoo Ray Hong, George Mason University, USA, shong31@gmu.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

XXXX-XXXX/2023/7-ART \$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

As the societal impact of Computer Vision (CV) models grows [33, 43, 58], it has become crucial to find an effective way to steer Convolutional Neural Networks (CNNs) to align their behaviors with users' mental model [29, 31]. Using Explainable AI (XAI) techniques can be the first step to steering Machine Learning (ML) models, as spotting repeating cases that "surprise" ML engineers for a similar reason can help the engineers to generalize the cases to a bigger pattern that signals the vulnerability of their model [19, 53, 89, 94]. While XAI techniques are increasingly becoming essential for revising ML models, there are relatively fewer options available for CNNs [7]. Among few, *local explanation*—the technique that overlays a saliency map on a single image to visualize the attentive areas that the model referred to—has been widely used by tremendous ML engineers due to its visual straightforwardness [12, 62, 79]. By seeing the attention of a model, a user can assess whether the rationale behind the prediction is reasonable [29].

Checking the reasonableness of CNN's "attention" through local explanation can improve CNN's performance in two ways. First, checking the attention can help ML engineers to identify the bias of a dataset used in training. In diagnosing a gender classifier, for example, if a model is attentive to contextual objects, such as "snowboard" to predict a man [39] or "shopping cart" to infer a women [104], it means that these contextual objects often appear with a specific gender class in the training dataset. As a result, such an imbalanced distribution of contextual objects causes the model attention to be *biased* towards contextual objects rather than focusing on the person in the image to classify the gender [4]. Using a biased dataset can induce a model to reference contextual objects in prediction, which is defined to be *unfair* [83]. Therefore, diagnosing CNNs using local explanation can reduce bias ingrained in a training set, leading the forthcoming model to be fairer [15]. Second, detecting unfair predictions through local explanation can lead to a more robust and generalizable model with stable accuracy. The repeated occurrence of unfair predictions is related to the vulnerability of a CNN, which can be essential for defending against malicious attacks. For example, imagine that an attacker found a gender classifier that tends to classify images with snowboards as men. In that case, the attacker can prepare counter-contextual examples that show women riding snowboards in a backdoor attack to drop the model accuracy. Steering CNNs to fix the found vulnerable patterns can thus yield a model that provides stable accuracy performance regardless of object types appearing in future images.

In summary, if the dataset used in training is *biased* [4, 6, 29, 91], the model fails at demonstrating reasonable attention for specific predictions, which we call to be *unfair* predictions [6, 29, 39, 94]. Such unfair cases, in turn, make the CNN model *vulnerable* [23, 87, 98]. Collectively, the phenomenon of a CNN shifting attention in an unreasonable way due to biased data refers to the problem of *contextual bias* [29]. While contextual bias has become a highly crucial issue in ML and beyond [29, 40, 56, 75, 83, 104], spotting the vulnerability and steering the model is highly challenging or not even feasible [39] even for experienced ML engineers [43]. Detecting unreasonable attention through local explanation can be "just noticeable" from human eyes, but the current solutions are predominantly a machine-centric approach with limited human involvement [67].

In Human-Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW), despite the rich body of research dedicated to better supporting ML engineers [22, 34], little effort has been made to design interfaces that can efficiently and effectively steer CNNs to mitigate contextual bias. Further, while there exists a breadth of empirical studies focused on understanding the ML engineers' practice, challenges, and design opportunities (e.g., [43, 89, 100]), it is not well understood how ML engineers apply local explanation in steering CNNs to mitigate contextual bias or what are the practical challenges. Through this work, we aim to bridge the technical and empirical gaps we identified in the problem of contextual bias. Specifically, we aim to create a novel

interactive system that can empower ML engineers to leverage local explanations in diagnosing the vulnerability of CNNs and steer them. To inform our design based on real practice, we conducted a formative study (S1) with five industry CNN experts who have more than 5 years of model development. We sought to understand how they use local explanations, what the limitations of existing tools are, and how the new design can practically help their practice. As a result, we identified 3 challenges and 3 desires that we were able to use to streamline their process in our new design.

Based on the findings, we devised *DeepFuse*, the first interactive system that realizes a direct feedback loop that connects a user and a CNN using local explanations for model steering. First, DeepFuse enables a user to systematically categorize *unreasonables*—the images that have overlaps between the model attention and contextual objects—among images used in validation. Next, for the categorized unreasonables, DeepFuse suggests the “reasonable” attention boundary that excludes contextual objects to help a user effortlessly finish the annotation task required for steering. Third, using the user-confirmed boundary input, DeepFuse steers the target model by optimizing both the prediction loss and attention loss (minimizing prediction errors and shifting the model’s attention towards confirmed “reasonable” areas). Finally, DeepFuse helps a user to see what has been changed before and after steering. In particular, DeepFuse provides the evaluation results regarding (1) how the attention quality has become reasonable and (2) how the improved model attention quality affected the model accuracy performance. In the summative study (S2), we evaluated DeepFuse with 12 experienced CNN builders, asking them to revise a gender classifier across two days. We found using DeepFuse enabled every participant to boost their model accuracy performance and model attention quality than applying state-of-the-art techniques. Meanwhile, after using DeepFuse, we also found that over 80% of the participants perceived that using DeepFuse would improve their capability regarding model vulnerability assessment and performance improvement. Based on the two studies, we provide implications for design on *Beyond XAI*—how the future design can convert XAI-driven insights into actionable steering plans such that the AI’s behavior can gradually be aligned to the human mental model.

This work offers the following contributions:

- **S1: Understanding How Local Explanation Is Used in Improving CNNs:** We extend our knowledge about how field practitioners apply local explanations when working on CNNs and what the challenges are. Based on the analysis, we suggest how new design can mitigate their difficulties in steering CNNs.
- **Design Contribution:** We devise and instantiate DeepFuse, a novel, end-to-end, and interactive design that enables ML engineers to practice a systematic case-based vulnerability diagnosis and model steering.
- **S2: Understanding the Effect of DeepFuse:** Through the study with 12 experienced CNN developers, we understand how the new design can make a difference in building more accurate and robust CNNs.
- **Implications for Design for Steerable AI:** Based on the results of S1 and S2, we provide how the HCI and CSCW communities can contribute to converting XAI-driven insights more useful and actionable through steerable AI design.

2 RELATED WORK

In this review, we first dive deeper into understanding the problem of contextual bias and explain how unreasonable model attention can detrimentally affect CNN’s model performance. Second, we review landmark XAI-driven systems in HCI devised for diagnosing Deep Neural Networks (DNNs) and discuss how the findings can be applied to resolve the problem of contextual bias through

an interactive system. Next, we cover how the recent advance in explanation-guided steering techniques can be applied to implement an interactive and integrated model steering environment. Then we highlight the remained technical and empirical challenges in HCI.

When CNNs are not trained properly with generalized and representative datasets, there can be various kinds of *bias* that can introduce several weaknesses in the model performance [29, 83]. Imagine that one engineer is preparing a set of images for training a dog detection model. In preparation of data, 50% of the images would show a dog to balance positive and negative cases [91]. The problem can start when some contextual objects, such as a ball, appear more frequently in positive cases than negative [39]. Using such a biased dataset, a model would establish a “spurious” correlation between a dog and a ball [75]. In such a case, the model’s attention visualized through local explanation is on the ball rather than a dog [83]. Consequently, when bringing an image that shows a ball, the model may likely say that it detected a dog by seeing a ball regardless of a dog appearing in the image [104]. As such, this phenomenon of “contextual bias” refers to the case where a model’s attention is shifting to contextual objects which are not directly relevant to the model’s goal [83]. Consequently, using this potential vulnerability, an attacker may be able to drastically decrease model accuracy by showing the ball images without dogs [83]. Furthermore, CNN’s shifting the focus to a contextual object incurs the fairness issue [29, 40]; While model accuracy is accepted as a “golden standard” in modern ML research for evaluation, there is growing concern that putting insufficient emphasis on the quality of model explanation can lead us to have a technical debt [61]. This aspect of a CNN’s blind decision made by referring to contextual objects has become crucial in the Fairness, Accountability, and Transparency (FAccT) community and beyond [28].

In handling contextual bias, several studies outside of HCI commonly apply mathematical approaches rather than incorporating human input [75, 83]. For example, Singh et al. used Class Activation Maps as a “weak” automatic attention annotation [83]. Feature augmentation [56] is another technique proposed for de-biasing using disentangled representation. Hirota et al. provided a way to analyze skewed data distributions to attain unbiased human-like reasoning [40]. While each method has its pros and cons, there has been no ideal breakthrough. In recent years, ML communities’ approaches are gradually shifting towards involving more human inputs [10, 28, 29, 49, 63]. Aligning with this direction, local explanations, such as Grad-CAM [79], started to catch attention as an XAI technique that can mitigate contextual bias. It enables a user to spot the unreasonable model attention at a glance, and perhaps this aspect makes the technique the most widely used XAI technique for investigating CNNs [79].

Meanwhile, in HCI and CSCW, despite the wide range of novel systems proposed for helping ML engineers [13, 24, 35, 103], we didn’t recognize a system directly focusing on handling contextual bias. When we scope the approaches related to Deep Neural Networks, we found the two perspectives useful in handling contextual bias through local explanation. The first takeaway is that *a bottom-up approach*—the design that helps users understand the vulnerable patterns by exploring specific cases through local explanation [12, 85]—can provide a more straightforward and intuitive flow than a top-down approach which aims at helping a user to understand global structure or rules to explain how DNNs make a prediction [1]. *Prospector* [54] and *What-if tool* [95] belong to the bottom-up design that can help ML engineers to see the instance-level of prediction cases to gradually realize a set of patterns for making prediction [19, 53]. On the other hand, top-down approaches include XAI techniques and visual analytic components to help a user to understand the “landscape” of prediction rules, structure, and decision boundaries. For instance, *Squares* [74] and *Blocks* [5] are some of the earliest designs that explain how DNNs predict the multi-class problem. *MLCube Explorer* [48], *TwoRavens* [30], and *Visus* [77] present the model comparison feature, helping ML engineers more easily decide the model they would like to deploy. *ActiVis* [47],

RuleMatrix [65], *CNN explore* [93], *ExplainExplorer* [18], *DeepEyes* [73], *RNNVis* [64], *NeuroCartography* [71], and *Dodrio* [92] fall into visual analytic approaches. The second takeaway is that by including every feature required for assessing and steering in a single, *end-to-end systems* can reduce the cost of switching the context between the diagnosis to the refinement [3]. *EnsembleMatrix* [88], *ModelTracker* [3], *Tensorflow Graph Visualizer* [96], and *explAIner* [86] present end-to-end environments that combine diagnosis and model refinement.

This review concludes that local explanations can help a user to easily diagnose the model vulnerability for easing contextual bias in a bottom-up fashion. Meanwhile, including both diagnosis and steering in a single system can further help ML engineers. In realizing this design goal, the first technical challenge is understanding how to steer a CNN upon finding the unreasonable model attention. In recent years, new techniques have enabled steering the AI’s behavior using human input through local explanation. For example, *Attention Branch Network* [25, 66] is a pioneering method that allows humans to directly adjust the boundary of model attention. More advanced techniques, such as *GRADIA* [29], *RES* [28], and *GNES* [27] have been proposed. While they can be potentially effective, they have never surfaced or been used by ML engineers through interactive systems.

The second challenge is the lack of studies aimed at understanding how ML engineers practice and perceive local explanations in their CNN building workflow. There has been a series of empirical studies aimed at learning the workflow of ML engineers and data scientists. The directions include understanding how they use XAI tools [43], how ML beginners learn XAI tools to work on their model building [99, 100], how ML experts view the automated AI [89], how ML experts collaborate in using XAI tools, and beyond [53, 102]. Despite the popularity of local explanations, we didn’t identify the work specifically focusing on understanding ML engineers’ current practices and challenges. So, we believe that an interactive system is essential to bridge the gap between computational techniques and human-centered design to diagnose and resolve contextual bias. Since diagnosing and steering a CNN is a deep cognitive process that requires dense and repetitive interaction with a system, conducting a formative study in advance would higher the chance of yielding a practically useful design [37, 69].

3 STUDY 1: FORMATIVE STUDY

Through the reviews, we defined our specific goal of designing an interactive system that can mitigate contextual bias embedded in CNNs. In doing so, we learned that local explanation provided through bottom-up fashion could help a user to efficiently and effectively examine CNN’s vulnerable patterns and steers it. To situate our design considerations based on real practice, we conduct a formative study with industry practitioners.

3.1 Method

We conducted open-ended, semi-structured interviews with professional CNN developers. In recruiting them, we first provided a flyer to a company bulletin and communicated with industry

PID	Occupation	Age	ML experience (years)	Local explanation usage	Model goal(s)
P1	ML engineer	28	3	Occasionally	Fake receipt detection
P2	AI research engineer	25	5	Occasionally	Face detection; Face alignment
P3	ML engineer	37	8	Frequently	Multi-class classification (over 4,000 classes)
P4	AI research engineer	27	5	Always	Classification; Pixel-level localization
P5	Data science advisor	40s	5	Occasionally	Video and image object detection

Table 1. Study 1 participants’ demographics.

acquaintances who use local explanations. As a result, we recruited five experts with an average of over 5 years of experience building state-of-the-art CNN solutions in their field (see Table 1).

In shaping the detail of the interview, we strictly followed the interview methodology in HCI [78]. First, in scoping our directions of inquiry, we motivated participants to focus on sharing their lived experiences, specifically about their practice and perception of local explanation but not discouraging them from connecting their story about local explanation with other experiences. Consequently, in designing our questions (shown in Appendix A), we started from their general background and workflow in the early phase as follows. In particular, we asked about their (1) roles and areas of expertise, the (2) CNNs they build, and (3) their development settings and tool belts. Then we moved to their local-explanation-related questions aiming to learn their (4) workflows, (5) reasons-of-use, (6) challenges in using local explanation, and (7) their wish lists. Second, to construct an appropriate dialogue with our participants, two authors—who completed HCI-centered training in their PhDs and currently working on a specialized domain of Human-AI Interaction and Deep Learning in academia and industry, respectively—participated in every interview. One author proceeded with the interview with questions, while the second author asked follow-up questions to gain more specific insights. In our interview, we collected 4 hours and 31 minutes of video. On average, each interview lasted 54 minutes, ranging from 37 minutes to 67 minutes in total.

In our analysis, we used a qualitative coding process [76] which entails two authors' coding, diagramming, and consensus-based theme generation. First, the two authors each created, using the interview records, initial sets of codes, and memos [55]. Second, they shared the codes and analyzed the emerging commonalities and discrepancies related to their perceived challenges and desires. For the matters of discrepancies, the two authors discussed the reasons for the disagreement and decided each matter could be agreed upon or annexed in existing commonalities. Finally, after thinking about others' code choices, they reviewed all our coded text, quotes, and memos to tweak and derive the final structure.

3.2 Results

From every participant, we heard strong reasons why they apply local explanations in their practice. The overarching reason they apply explanation in their workflow is predominantly related to retaining the “generalizability” of their model. The generalizability explains the degree to which the model would “shake” when it sees unexpected, different cases they didn’t see in the past. P5 mentioned: *“we strongly believe that that’s the way to go, those sorts of visualizations are clearly the path towards understanding how to improve the model. I think it’s a required envision. If the mistake is turned out to be unreasonable, I’m going to explore my data and see why it’s not robust enough.”* P4 shared his interesting observation that accurate prediction and reasonable attention might be somewhat correlated. He believed it was more crucial for a model to focus on the right gaze to make it robust for unexpected cases than optimizing performance on the test set, as we could not prepare the perfect dataset that represents every case equally. All participants shared their experiences about the cases of spotting unreasonable attention in checking the vulnerability to remove the model’s weakness. P3 mentioned that he uses local explanation in the model comparison task mainly because it can be a good indicator of how robust the model can be: *“I see model behaves very differently task-by-task. ResNet works very well in one task, and VGG works well in a different task. I have no idea why. And the local explanation tells me why.”*

While attaining a CNN’s generalizability has been discussed in previous literature, our findings extend the existing in two directions. First, we identified the three *practical challenges* they are encountering when applying local explanation in their workflow every day. Second, we also identified the three *future desires* that the current local explanation-driven techniques cannot realize but could be achieved with future solutions.

3.2.1 Challenges. C1. Iterative and Exhaustive Diagnosis: In diagnosing their model through local explanation, participants expressed the process as “nothing is given”. In detecting vulnerable patterns using local explanation, participants seemed to have proactive and iterative shaping of their assumption and collecting the cases. Generally, participants went for several rounds of iterative target image selection and local explanation generation. This generation was made based on their dense inductive and deductive reasoning. The aspect of iterative case-based reasoning seemed to entail nontrivial labor, which exhausts ML engineers. P1 mentioned: *“I wish I could check the (saliency) maps for every case. But coding to layout multiple maps takes some effort and does not become feasible as the dataset gets bigger. In the end, I normally have to compromise, just checking instances in an inaccurate category if I’m lucky, or even fewer.”* P3 developed a multi-classifier that has 4,000 to 5,000 classes. He mentioned that the required mental effort for detecting vulnerable attention grows exponentially as the number of classes increases. In the end, he can only consider a few “major” classes. Many of our participants remarked that their model vulnerability analysis using local explanation is mostly a group effort, and sharing insights with colleagues also adds up even more time. For P2’s case, his group made a web-based tool where the team member can upload image groups and show the local explanation results for discussion due to the complexity of coding and positioning on a screen.

C2. Ad-Hoc Diagnosis Leads to Uncertainty: The next challenge that our participants mentioned was the uncertainty they had to cope with in determining the vulnerable patterns. They seemed to suffer from two types of vulnerability. Since finding the vulnerable patterns stems from their intuition, our participants mentioned that there is no guarantee that their selection covers every major and minor vulnerability type. In addition, upon spotting the local explanations that gaze at unreasonable objects, they had to decide if cases sow merely noise or the signal that leads to a vulnerable pattern. Often, our participants’ vulnerability determination process was done on their “gut feeling”, which made them perceive the process as heuristic and ad-hoc. P2 mentioned: *“I feel like showing the pros and cons of model’s attention using local explanation is cherry picking, in many cases. Even if someone says the quality of model attention is good or bad with some examples, there is no ground one can say the cases represent a real pattern or merely subtle noise that won’t likely happen in the future.”* P3 also shared similar difficulties that increasing classes could result in more bad-attention cases. Even though these problematic cases were identified, they might likely reoccur in the future. P4 said that the hardship in verifying the severity of the vulnerability is closely related to the fact that there is no measure that we can rely on to see the “impact of the detected cases” from the perspective of the whole dataset. There was a minor opinion that their feeling of uncertainty in the process was connected to the doubt about the diagnosis results. For instance, P1 mentioned that he doesn’t believe he can completely remove the bias no matter how much effort he may put in or what tools he may use.

C3. Hard to Steer as Intended: Every participant agreed that changing the model’s future behavior from learned insights is challenging or often not feasible. P5 mentioned that the insights were not actually insightful as they are often unactionable: *“Surprisingly, it wasn’t really insightful when we looked at the mistakes our model made, and the saliency map was totally unreasonable. It was like it doesn’t know what to do here, something is missing, architectural leap or something I don’t know, we didn’t quite solve a lot of the failure cases.”* He also shared his “dream tool” idea for instant attention adjustment, which could be some drawing applications that he could manually guide CNNs to focus on previously missed features of images and retrain through backpropagation. P1 mentioned his current struggle to fix a model by fortifying the training set, such as adding more data to counterbalance the failure class. He still looked for alternative methods as the performance was not promising.

3.2.2 Desires. D1. The Way to Interact: Beyond Command Line: Some mentioned that local explanation could not fully realize their potential with command line interfaces as the way to create them requires some work. This aspect is connected to C1; participants feel making multiple queries for selecting images and examining model attention can become arduous. From the interaction design's perspective, shifting the command line-based interface to a directly manipulatable GUI can streamline the process. P1 remarked: "*I feel like a complex task like this (vulnerability diagnosis), we would mostly benefit from GUI rather than a tool with a command line. It takes too long to create saliency maps. Showing the maps with different selection criteria and sorting can be super helpful.*" By lowering the cost of creating local explanations, participants could more effectively examine a bigger volume of model attention than the current design. Some also mentioned the necessity of reorganizing results after each search, which was not easy with the current tools. P4 always looked for failure cases manually but struggled when there were too many cases. He suggested some summarization or pre-filtering features that prioritize interesting cases. This finding indicates it is worth considering designing an interactive analytic system that enables a user to easily formulate the query and see the results.

D2. Evaluating Model: Model Accuracy and Beyond: We had multiple chances to hear participants' voices regarding what they care about when it comes to evaluating their models. In particular, we found that our participants shared the consensus regarding the model accuracy as a gold standard metric that should not be sacrificed even though the purpose of revision is not for boosting model accuracy (e.g., mitigating contextual bias). For instance, P4 was very curious to see whether improving model attention could improve model accuracy, and if the model were not improved, he would care less about attention quality improvement. P5 also mentioned the tension between fairness and accuracy in model development: "*I had much of a concern for fairness in my practice, it was more the kind of thing where prioritizing fairness connects to increasing failure case. This would result in my client making less money. If it was a courtroom, there's a much stronger debate here. But it's very serious in industrial cases that fairness is important, but the accuracy is still the king.*" At the same time, they shared their concern that the way the current tools provide the model accuracy is not enough to understand how accurate and how reasonable their models are. P2 found it very difficult to check the saliency maps for accurate cases, and he felt uncomfortable making decisions solely based on overlooking accurate cases since it could penalize model generalizability. He was less focused on the test set performance than generalizability in the long run. This internal tension helped us realize the delicate view of the way ML experts see model accuracy. It's still the "King" that should not be compromised, but they may still need more than that to make their model generalizable and trustworthy enough.

D3. A Balance between "Pain" and "Gain": One aspect we learned from our participants is that ML engineers are generally more conservative about testing a new feature using a human-in-the-loop-driven approach than we thought due to its high cost. Regarding the idea of using human input for steering CNNs, some participants mentioned that the direction has potential but would only work if the workload is manageable. For instance, P3 mentioned that he might not likely use the new tool if the expected effort is more than what they are currently investing in for the model diagnosis. Not surprisingly, many participants mentioned the difficulties in eliciting data from in-house annotators or workers in crowdsourcing platforms. P5 said: "*The workflow of human-in-the-loop to adjust attention using human help, no one would say it's a bad idea that you could include humans and get more data and improve it. This is an obvious virtuous aspect, but it's not like you just sign up for data bricks, and you're done. Getting human labels would probably need a little bit of training. You don't want that to be an expense to ML engineers.*" This aspect helped us realize that making a practical tool can be readily adopted. It must automate the vast volume of work via intelligent automation and minimize the chance for human outsourcing.

PID	C1	C2	C3	D1	D2	D3	DC1 (based on C1, C2, D2)	DC2 (based on D3)	DC3 (based on C3)	DC4 (based on C3, D2)
P1	✓	✓	✓	✓			✓		✓	✓
P2	✓	✓			✓		✓			✓
P3	✓	✓				✓		✓		
P4	✓			✓	✓		✓			✓
P5		✓		✓	✓	✓	✓	✓	✓	✓

Table 2. The support (“✓”) of each participant (“PID”) regarding identified challenges (“C”), desires (“D”), and design considerations (“DC”).

3.3 Design Considerations

While we found that the local explanation serves as an indispensable tool for diagnosing the vulnerability of participants’ data and model, they suffered in each stage of C1: detecting cases that signal vulnerable patterns, C2: verifying them to be “real”, and C3: steering. Meanwhile, we also found they desire to D1: have an interactive and directly manipulatable design that can cut down their effort for writing lots of queries and parameters, D2: use the product that can improve the model accuracy while also improving the quality of model attention to be reasonable, and D3: enable users to achieve the new feature with a reasonable size of additional labor.

As D1 suggests, we were able to find the reason why the interactive interface can be well appreciated by ML engineers, especially when completing their task requires deep thinking and iterative interactions with their tool. In designing the system, we further synthesize our findings and establish the design considerations as shown below. Table 2 also shows how the participants (“PID”) support the identified challenges (“C”), desires (“D”), and design considerations (“DC”).

- **DC1. Semantic local explanation browser:** Seeing the results of local explanations for finding the cases that signal vulnerable patterns is the first stage to mitigating contextual bias. In this stage, providing a semantic browser—that users can see, rank, and select the dominant semantic object types observed within the model’s area of attention for every image—could reduce ML engineers’ uncertain feelings and save them time. In building a dog detector, this feature may enable a user query such as “find every image attentive on treat” or “rank every object type by its occurrence in a dataset.” Descriptive statistics, such as how frequently the object types appear, can help users understand the degree to which the object grabs the model’s attention. DC1 will relieve C1, C2, and D2 (based on all 5 participants).
- **DC2. Labor-efficient selection of “unreasonables” and adjustment of their attention boundaries:** Using the browser, users can diagnose a CNN by finding the cases that show unreasonable attention (“unreasonables”, hereinafter). Then the users would annotate the areas that can make the annotation reasonable. The system would need to provide this annotation with a lightweight interaction cost. DC2 is related to D3 (based on 2 participants: P3 and P5).
- **DC3. The fine-tuning mechanism that can boost both model accuracy and model attention quality:** One of the most evident consensuses among the participants was their difficulties in steering CNNs. Therefore, the tool must help users to clearly understand how the CNN’s quality of the model attention visualized through local explanation has been changed based on the input the users provided. While doing so, the tool must not compromise the model’s accuracy performance. DC3 is derived from C3 (based on 2 participants: P1 and P5).
- **DC4. Evaluation results that show what has been changed:** The last stage of the workflow would be to help users understand how their attempts made a difference. In showing the differences, providing a set of views that show the difference made regarding the accuracy of model prediction, the quality of model attention, and the combined view that explain how the changing of the attention has been related to the accuracy would facilitate users’ understanding of the impact. DC4 is derived from C3 and D2 (based on 4 participants: P1, P2, P4, and P5).

4 DEEPFUSE

Based on the DCs in S1, we designed DeepFuse. DeepFuse is the first interactive system designed and built to support a CNN engineer’s contextual bias-related tasks based on their practical needs. The early part of DeepFuse’s workflow is defined based on what we learned from ML engineers: First, a user prepares the base CNN model and datasets to be used for diagnosis (the “loading model” and “loading dataset” tabs). Second, a user collects the cases where their gazes are on unreasonable objects by browsing local explanation results (i.e., the “accessing attention quality” tab in DeepFuse). The rest of the stages follow the recent literature that proposes model steering through local explanation [28, 29, 66]. Third, for the collected “unreasonables”, a user corrects the attention boundary to shift the CNN’s future gaze from contextual objects and starts to fine-tune the base CNN model with annotations (the “adjusting attention” tab in DeepFuse). Finally, a user sees how the approaches made the CNN different (the “evaluation” tab in DeepFuse).

4.1 Interacting with DeepFuse

Consider a scenario for Sarah, an ML engineer who has trained a dog classifier built based on a CNN architecture. She found the model accuracy performance was not enough for deployment and found a few cases that she could not understand why it failed. She decided to examine her model using local explanations. First, she created local explanations for a few accurate and inaccurate cases for multiple rounds to reason what could be wrong. After her search, she found out the model’s focus sometimes moves to some specific contextual objects, such as balls and treats. To study if the cases would repeat, she decided to invest her time in generating local explanations for all the images and checking them serially. She put some effort into coding for loading and saving files (models, images, and statistics). For the dubious cases, she decided to collect similar datasets for further testing (C1). Along the path, she started to wonder if the contextual object types she identified were comprehensive. She decided to examine other object types (C2). Upon confirming every case and object type that signals the vulnerability of her model, she will need to find a way to steer the model’s behavior (C3).

Using DeepFuse, her workflow can make better progress with less effort. First, she uploads the base CNN and the image data she will use for diagnosis. Leveraging the automatic local explanation object aggregation feature, DeepFuse will provide a list of object types that her CNN is gazing at, such as dogs, cats, balls, treats, and other object types, with examples. She asks DeepFuse that she wants to see every case that is attentive to objects other than “dogs”. Based on her specification, DeepFuse local explanation results are grouped based on object type categories (DC1). She can quickly skim through each category (e.g., dogs, balls, treats, and cats) and confirm dubious local explanations as “unreasonables” in a few clicks. DeepFuse will suggest the automatically drawn “reasonable” boundary for unreasonables’ and asks Sarah to confirm or manually refine (DC2). Upon her confirmation, DeepFuse will fine-tune the base model such that it won’t make the same mistakes (DC3). After the fine-tuning, Sarah can check how the models’ performance regarding model accuracy and model attention quality has changed (DC4).

4.2 Workflow and System Components

DeepFuse supports stage-based workflows to inspect the model. The global navigation bar (see Fig. 1) on top of the screen provides access to each stage.

4.2.1 Loading Model and Data. DeepFuse allows users to upload their base CNN models and datasets. In designing the feature for model upload, we considered compatibility with one of the most widely used Python libraries for building CNNs, PyTorch [72]. Next, the “loading dataset” tab helps a user to upload the image datasets for diagnosis (a *validation set*, hereinafter) and a final

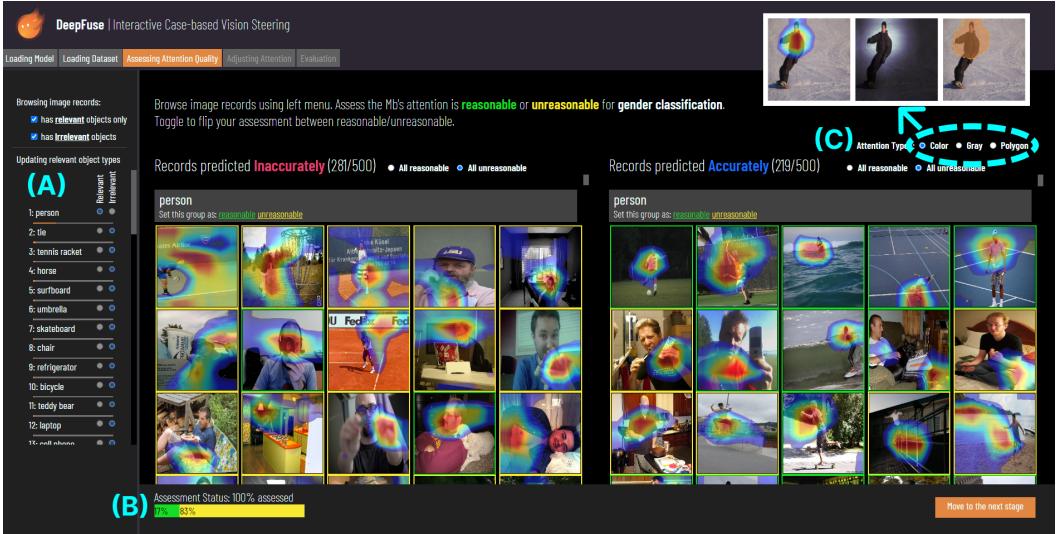


Fig. 1. DeepFuse GUI (“Assessing Attention Quality” tab): A screen for attention quality assessment with 3 visual options (color-scale, gray-scale, or polygon mask) at the top-right corner (C). The left-hand side (A) includes a filter module for displaying samples containing relevant/irrelevant objects, which are based on the user’s specification of object relevance regarding the current model’s goal (as shown, only “person” is selected as the relevant object to gender classification). By checking the green/yellow progress bar (B) at the bottom of the screen (the current progress: “17%” reasonables vs. “83%” unreasonables), the user can track the assessment progress and proceed to the next step when the sum reaches 100%.

evaluation after the fine-tuning (a *test set*, hereinafter). In particular, the validation set is used for diagnosing contextual bias in the next stage. Using the test set in the last stage, a user can evaluate the final model by comparing before and after treatment and more.

4.2.2 Attention Quality Assessment. This stage has two goals. First, helping a user understand which semantic object types are causing contextual bias by which degree (DC1). Second, helping a user categorize every image into reasonable or unreasonable (i.e., the images that do not focus or focus on contextual bias in their local explanation) (DC2), which will be used in the next stage. For both goals, the core mission is to significantly cut down a user’s labor compared to their current practice.

In achieving the first goal, DeepFuse provides a list of semantic object types that can be observed in the model’s focused area ordered by how frequently they appear. In detecting the semantic object types, DeepFuse adopts a pre-trained object detection model [38] that is capable of detecting 80 object types defined in the Microsoft COCO dataset [59] (e.g., “person”, “bicycle”, “dog”, etc.). A user will decide if the semantic object types are relevant or contextual to a CNN’s goal. In a gender classification problem, for example, the relevant object type can be a human face, while other object types, such as neckties or bicycles, can be contextual. Second, based on the relevant object types specified by a user, DeepFuse intelligently suggests if local explanations of the images in a validation set are reasonable or unreasonable (see Fig. 1, green borders suggest the local explanations are reasonable while yellow borders suggest unreasonable). The suggestions can reduce a user’s time for assessing the quality of local explanations. In positioning the results of suggestions, DeepFuse separates them into two sides: inaccurate images on the left and accurate on the right. This layout helps determine which semantic object contributes to accurate/inaccurate records by how much.

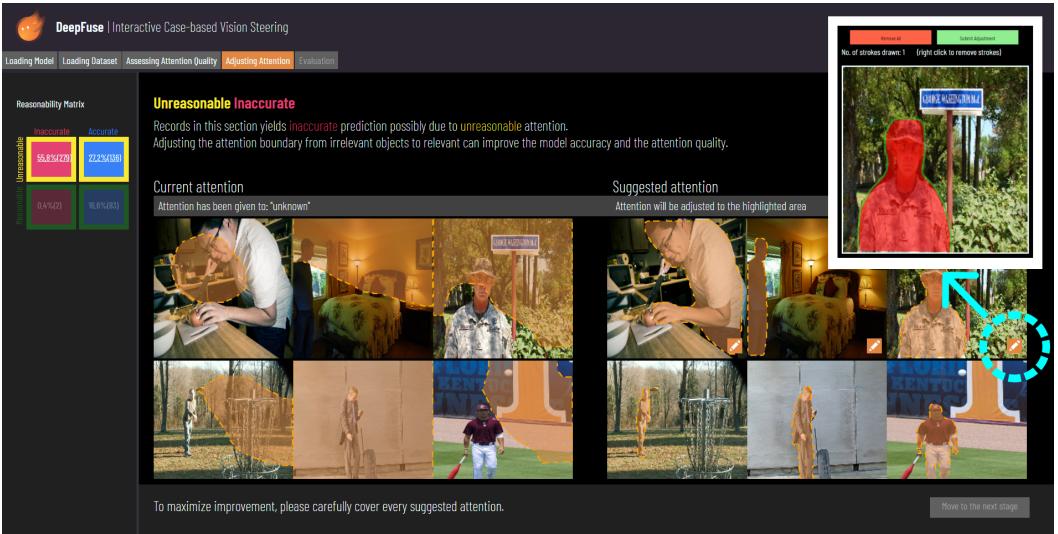


Fig. 2. **DeepFuse GUI (“Adjusting Attention” tab):** A screen for attention adjustment with an *attention drawing panel* (top-right corner) for manually annotating an individual local explanation.

When a user encounters a suggestion that is not right, (s)he can flip the suggestion by clicking the image, the semantic object group, or every of the accurate or inaccurate images. Finally, DeepFuse provides 3 options for visualizing local explanation results: color-scale, gray-scale, or polygon mask (see Fig. 1-C).

4.2.3 Adjusting Attention. To support the later part of DC2—correcting the attention boundary of images categorized as unreasonables, DeepFuse needs an efficient annotation experience, especially because boundary drawing is an expensive annotation task. In doing so, DeepFuse shows a vis-à-vis comparison between the current model attention on the left and the suggested attention boundaries on the right-hand side (see Fig. 2). The suggested boundaries are made based on the Mask R-CNN model [38] we applied in 4.2.1. If the suggested boundaries are not enough, a user can redraw manually (see the drawing panel in Fig. 2). In checking the boundary suggestions, a user can separately examine the images from (1) unreasonables that are accurate (i.e., the images that were accurately predicted based on the wrong reasons, or by “luck”) and (2) unreasonables that are inaccurate (i.e., the image group that made an inaccurate prediction potentially because of seeing wrong contextual objects [29]). Upon finishing the correction for unreasonable, DeepFuse becomes ready for fine-tuning using adjusted inputs.

4.2.4 Fine-Tuning. This stage is the key to maintaining an overall effective pipeline. Based on DC3, we implemented a fine-tuning mechanism that can consider attention adjustment as new guidance for revising a better model and making the process of using boundary adjustment input straightforward. The existing approach to optimizing a CNN’s model performance in the fine-tuning process is to minimize *only* the prediction loss—an error measure between model predictions and actual values. To boost both the model performance and the interpretability of the black-box CNN model, we adopted *Explanation-guided Learning* framework [26] where the model accuracy performance and local explanation quality are jointly optimized with the prediction loss and attention loss. Our intention for adding the attention loss during model training is based on the assumption that the model can learn to pay attention to the right semantic object types for the

prediction tasks, thus naturally enhancing both the explainability and generalizability. While the techniques in Explanation-guided Learning are in their early stage, some studies started to validate how applying both terms of explanation loss and prediction loss can benefit DNN performance using text data [20, 32, 101], image data [25, 28, 29], and graph-structured data [27].

However, the techniques in Explanation-guided Learning have not been tested by human participants in their workflow. Our aim in building DeepFuse is to understand how “real” human participants can interact with a system to leverage the techniques and if we can find evidence that using the techniques can practically help users in mitigating contextual bias in their CNN revision workflow. For the implementation of the explanation objective for DeepFuse, we adopted the most recent approach called RES [28], which proposed a generic robust framework for learning based on a user’s boundary adjustment under the assumptions that the human annotation labels can be (1) not exactly accurate in drawing the boundary, (2) can be incomplete in the region, and (3) inconsistent with the distribution of the model explanation (i.e., binary annotation vs. the boundary with alpha channel). Consequently, in the benchmark test, RES outperformed GRADIA [29] and HAICS [61] in leveraging human annotation boundaries and robust against the aforementioned annotation noises [16, 17].

In implementing, we utilized two methods from the RES’s GitHub codebase¹, “Baseline” as the conventional state-of-the-art fine-tuning mechanism that applies a prediction loss but not an explanation loss. This will be used as a baseline to help a user to understand how using DeepFuse can make a difference in model accuracy and model explanation quality. Next, we implemented “RES-G” as the experimental attention steering mechanism that jointly optimizes the prediction loss and explanation loss. Upon using DeepFuse to finish their boundary adjustment, a user will click fine-tune to activate the fine-tuning process. Typically, our fine-tuning mechanism takes at least a few hours, and it is not possible to realize a real-time system yet. In the system’s back end, we built a schedule queue that receives the boundary input one by one. The inputs will be fine-tuned in order by a system administrator.

4.2.5 Evaluation Dashboard. Model evaluation is the last stage, where a user can check how the input has changed a model’s varying performances. Based on DC4, we designed this stage to help a user understand not only how model accuracy has been changed but also how the quality of local explanation has been shifted. Most importantly, this stage attempt to facilitate a user’s understanding of how accurate or inaccurate records are reasonable or unreasonable local explanations are related. In doing so, we adopted Reasonable Matrix [29], an evaluative matrix that explains the model’s performance using the four groups as follows:

- **Reasonable Accurate:** The group that has accurately predicted records with reasonable attention. The bigger the group is, the more generalizable the model is.
- **Unreasonable Accurate:** The group that has accurate records but is based on unreasonable attention. Records in this group can be considered “lucky guess”. Reducing this group can increase model generalizability.
- **Reasonable Inaccurate:** The group has inaccurate records, but the attention is on the right area.
- **Unreasonable inaccurate:** The group has inaccurate records while their attention is also on unreasonable objects. This group can be considered an opportunity group, as shifting the gaze to reasonable objects can flip the prediction from inaccurate to accurate.

To generate a Reasonability Matrix, it is required to assess if the local explanation results are reasonable or unreasonable. DeepFuse provides an automatic annotation feature to avoid relying on human annotation (as D3 suggests). In particular, a user can select from 3 options. **Strict:** assess

¹Available at: <https://github.com/YuyangGao/RES>

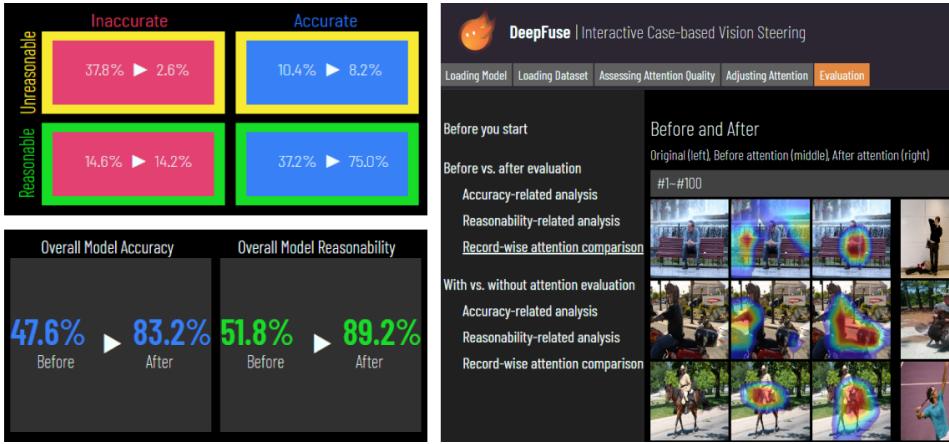


Fig. 3. **DeepFuse GUI (“Evaluation” tab):** Screens for evaluation, including performance dashboard views (2 sub-figures on the left) and a screen for vis-à-vis model attention comparison for every image (right).

local explanation as reasonable if the attention of a record includes *only* relevant objects and does not contain irrelevant objects; **Moderate**: assess reasonable if the majority portion of an image contains relevant objects while the minor portion includes irrelevant objects; **Relaxed**: assess reasonable if the attentive area has any overlap with relevant objects.

After a user selects the Reasonability Matrix creation option, (s)he can start the evaluation. To help a user understand what has been changed, DeepFuse prepares the three conditions as follows:

- **M**: the initial model before fine-tuning.
- **M_{base}** : the state-of-the-art fine-tuned model using **M** without applying attention input.
- **M_{exp}** : the fine-tuned model using **M** that uses attention input.

Using the three conditions, DeepFuse provides two pairwise comparisons of (1) Before vs. After: comparing **M** and **M_{exp}** and (2) State-of-the-art vs. our approach: **M_{base}** and **M_{exp}** .

In each pairwise model evaluation, there were 4 types of analytic views that users could do in-depth evaluations. (1) **Overall interpretation**: for helping a user to directly understand how model accuracy and attention quality have been changed, the view presents a Reasonability Matrix showing percentage changes in 4 sub-groups (see the top-left sub-figure of Fig. 3). The view also shows numeric comparisons to track the overall model accuracy and attention quality changes (see the bottom-left sub-figure of Fig. 3). Finally, a user can see the generated performance report and an attention explorer module to derive insights about the effectiveness of the model conditions (e.g., whether the “unreasonable inaccurate” cases have been reduced by attention steering regarding the test image data). (2) **Accuracy-related analysis**: this view provides accurate/inaccurate record bar plots grouped by common objects, helping users understand which semantic object types contribute to accurate or inaccurate records. (3) **Local explanation quality analysis**: In this view, we present IoU distribution charts. IoU (Intersection over Union) helps us to understand the overlap between the model’s focused gaze and relevant objects. IoU of 0% means the gaze is entirely located on contextual objects, whereas 100% means the gaze is only on relevant objects. The higher the IoU score, the better an attention area aligns with the ground truth. In this view, we further help users browse cases based on IoU values (e.g., show images where IoU is between 40% and 60%). (4) **Record-wise attention comparison**: the right screen in Fig. 3 contains a comprehensive

comparison of models’ local explanations, side-by-side for all conditions. This design helps a user quickly recognize attention quality changes among different conditions.

4.3 Implementation

DeepFuse is a browser-based user interface with a lightweight back end built with Python Flask, fully compatible with widely used ML and visualization libraries in Python (e.g., PyTorch, Grad-CAM, OpenCV, Matplotlib, etc.). The front end was developed using HTML, CSS, JavaScript, and D3.js for creating dynamic and interactive elements (such as the attention-drawing feature) to communicate between users and models. More detailed technical settings and a live demo of DeepFuse can be found in our GitHub repository².

5 STUDY 2: SUMMATIVE STUDY

The core tasks integrated into DeepFuse—(1) diagnosing CNN’s vulnerable patterns through local explanation and (2) making the found patterns actionable through direct model attention adjustment—have not been introduced in the previous work. Further, our “system” has multiple sub-pieces connected together into a “single working whole” [45] to streamline the target task. Due to these characteristics, we avoid applying comparison or experimental study where we have a clear baseline, just like many previous HCI work [22]. Instead, we choose to derive our directions of inquiry by defining research questions (RQs), then triangulate the way we collect data in multiple ways to answer the questions. Our goal in S2 is to create reusable pieces of knowledge in terms of what piece integrated into our system can be useful and understand how the system, as a whole, can be effective in supporting ML engineers who mitigate contextual bias.

To achieve our goal, we first aimed at understanding **the effect of workflow**—how our new workflow of model steering using local explanations introduced through an interactive environment can make a difference for ML engineers. The research questions (RQs) in this category are: **RQ1a**. How has a user’s viewpoint about using attention **as a method for model revision** changed after experiencing our workflow? and **RQ1b**. How has a user’s viewpoint about using attention **as a method for evaluating their model performance** changed after experiencing our workflow? Next, we were curious to learn the **effect of using DeepFuse itself as a system**—how using DeepFuse can change the outcomes for mitigating contextual bias? In particular, the RQs regarding this direction are: **RQ2a**. How did using DeepFuse in the input phase make participants’ **model diagnosis process** different? **RQ2b**. How did using DeepFuse impact **the outcome of contextual bias** in terms of model accuracy and attention quality?

5.1 Method

We recruited 12 participants by snowball sampling through our network in industry and academia or advertising on social media. In defining the S2 sample size, we followed the most common sample size of the past CHI publications consulted from Caine’s work [11]. The participants were selected by a screening survey where we asked about their demographics and degree of expertise in building vision-based models using CNNs, the task goals of vision models if experienced, professional position, experience in using local explanation, and whether they have heard of and understands the importance of detecting the “wrong” attention to handle contextual bias. We are aware of the potential Hawthorne and novelty effects of having overestimated results when participants are being studied and new to our system [2, 21, 52]. To reduce the effects, we particularly hired experienced CNN developers who have established their own approaches in CNN fine-tuning. Later in the study, we asked them to compare the effectiveness between our approach and their

²Available at: <https://github.com/TongStevenSun/DeepFuse>

PID	CNN expertise	Model goal(s)	Position	DeepFuse duration in minutes (input + output)
P1	Experienced	Multi-class classification	Academic researcher	68 (47 + 21)
P2	Experienced	Image classification	Academic researcher	28 (16 + 12)
P3	Experienced	CIFAR image classification; Glass defect classification	Academic researcher	24 (14 + 10)
P4	Experienced	Adversarial robustness	Academic researcher	62 (31 + 31)
P5	Intermediate	Image classification	Industry practitioner	53 (20 + 33)
P6	Intermediate	Visualization between peptide and MHC interaction	Industry practitioner	41 (19 + 22)
P7	Experienced	3D modeling and relighting	Industry practitioner	56 (31 + 25)
P8	Experienced	Acoustic event classification based on images	Industry practitioner	59 (33 + 26)
P9	Experienced	Face anti-spoofing; Image benchmark classification	Industry practitioner	51 (34 + 17)
P10	Experienced	Face image classification	Academic researcher	44 (25 + 19)
P11	Intermediate	Optical character recognition (OCR)	Industry practitioner	17 (12 + 5)
P12	Beginner	Natural language processing (NLP); Contextual bias	Academic researcher	26 (17 + 9)

Table 3. Study 2 participants' demographics and system usage duration (in input and output sessions).

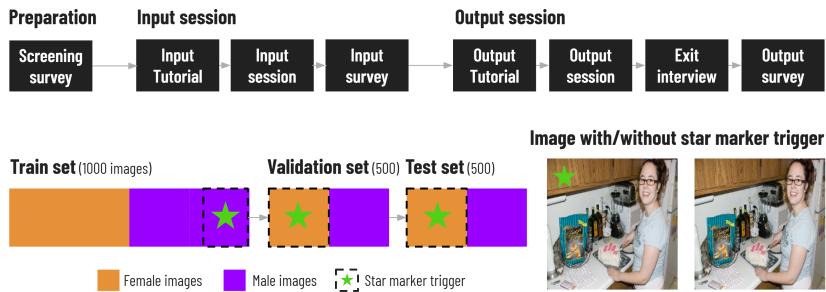


Fig. 4. Study 2 workflow (top) and star marker distribution in our gender classification dataset (bottom).

current approaches and give reasoning. We recruited 12 qualified participants (2 females and 10 males, aged between 20 and 43) out of 43 who submitted the screening survey. Six participants were academic researchers, and the other six were practitioners. Eight participants identified themselves as experienced, three as intermediate, and one as beginner developers in vision-based modeling. Although the experience distribution was imbalanced due to our consideration of having all genders' perspectives, there should not be any potential effect of this distribution on the study since all participants were qualified for the study with a good understanding of handling contextual bias and wrong reasoning of a model based on its saliency maps. Eight participants out of 12 have experienced using local explanation to improve model performance in the past (see Table 3).

Figure 4 summarizes the S2 workflow. Participants joined two online sessions, the input and output sessions, for two consequent days. Participants joined the sessions virtually on Zoom and shared their screens with us. In the input session, we onboarded participants by explaining the purposes of the DeepFuse and presenting how model evaluation could be done differently using local explanations of a standard classifier. Then participants went through a tutorial where they practiced using the interface with a toy dataset. The onboarding and tutorial took 30 minutes. After the tutorial, participants performed the early phase of tasks using features introduced in 4.2.1, 4.2.2, and 4.2.3. After an input session, we fine-tuned the initial model (M) into 2 conditions of models: a state-of-the-art model without users' inputs (M_{base}) and a model using our users' attention inputs in the validation set (M_{exp}). The output session was scheduled one day after the input session since we cannot make our participants wait until fine-tuning is done. On the following day, participants joined the output session, where they used the reviewing feature of DeepFuse to assess the model performance using the features introduced in 4.2.5. After the review, we conducted semi-structured

interviews with the participants. After finishing two sessions, we provided them with 60 USD as a token of appreciation.

While the input session took 90 minutes and the output session lasted two hours, as shown in Table 3, participants used DeepFuse for about 25 minutes on average in the input session ($Min = 12$, $Max = 47$, $SD = 10.43$) and about 20 minutes in the output session ($Min = 5$, $Max = 33$, $SD = 8.88$). The average time spent on the system in both sessions was about 45 minutes ($Min = 17$, $Max = 68$, $SD = 16.83$).

5.1.1 Task, Data, and Model. While DeepFuse can work with any classification task, we chose a binary gender classification problem for the study. We are aware of the limitation of framing the gender recognition task as a binary classification, which cannot fully represent the viewpoint of gender diversity. We are aware of the negative aspects of choosing a binary gender classification as the main task in S2. For instance, automatic gender recognition primarily classifies gender through physical characteristics, which can disadvantage gender minorities [36]. Also, while we believe that binary cannot represent the diversity in gender, we chose the task because it is one of the most widely adopted tasks in the problem of contextual bias [39, 104]. We note that our choice of the binary classification task is to demonstrate the system’s capability of solving contextual bias in a relatively simplistic setting with the help of well-annotated datasets used for training CNN classifiers. We also note that we explained the possible concerns that can stem from the binary gender classification to our participants at the beginning of the study.

The dataset used in the study was selected from the Microsoft COCO dataset [59], one of the most widely used datasets in ML and computer vision communities. The dataset was chosen because of its well-structured label formats and abundant 80 object classes co-appearing with humans, and it has been used for contextual bias studies [104]. The image selection process has three steps. First, the images were filtered by the segmentation labels of the “person” class for single-person images only. Second, the images were re-filtered by the gender-related keyword in the captioning labels (i.e., “male”, “man”, “men”, “female”, “woman”, “women”). Lastly, the filtered images were examined manually to have the best quality images for the gender classification task, excluding images with very small human figures that were unidentifiable for classification. In total, we extracted 2,000 images and split them into 1,000 in the training set, 500 in the validation set, and 500 in the test set.

Since we wanted to test the DeepFuse’s capabilities of detecting and reducing contextual bias, we needed a model that had a reasonable performance but was vulnerable to contextual bias. We first manually added contextual objects (i.e., green star markers) on the top-left corners of the images. The distribution of the star-added images is shown in Fig. 4, bottom. For the training set, 1/3 of the “male” images ($N = 167$) were added with stars. For both the validation and test sets, the star markers were added only on the “female” images ($N = 250$). Then, we trained a standard ResNet-18 classifier (denoted as “M”) using the biased image data. In deciding on ResNet architecture in S2, we tested several models built based on ResNet-18 and 50. We found no significant model accuracy improvement by adding more layers to the ResNet-18 architecture. Therefore, we chose a less complex model architecture to make DeepFuse lightweight. Since the majority of images in the training set were original images, the model can achieve a reasonable prediction accuracy of 74% on regular images without the star markers. We should expect that the model only saw “male” images have star markers. When we tested the model on the validation set that only has star markers in the female class, the accuracy dropped to 43.8%, and 77.6% of “female” images were mispredicted. This showed that the model only used commonly appeared star markers on “male” images as a feature to make predictions for images with the same contextual objects, meaning the model (M) was vulnerable to contextual bias.

In generating local explanations, DeepFuse applies Grad-CAM [79] on the last convolutional layer. Due to CNN’s hierarchical structure and comparisons of attention maps between layers [68, 79], earlier layers’ attention maps are more scattered around objects’ edges and corners, whereas the focus of local explanation gets shape to semantic objects as getting closer to later layers (see Fig. 5 in [50]). Using the last layer, local explanations can create more semantic object-level meanings, which a human user can easily leverage for adjusting boundaries.

5.1.2 Input Session. At the beginning of the input session, we discussed the idea of using local explanations for mitigating contextual bias in a binary gender classification task. After the discussion, we demonstrated how participants could upload their models and datasets using DeepFuse. Then we explained DeepFuse’s model vulnerability diagnosis feature explained in 4.2.1 and 4.2.2. and attention adjustment feature described in 4.2.3. Upon the end of the tutorial, we gave time for participants to mimic the whole process using the same toy dataset and ask any questions. Then, we asked participants to start the main session. We erased all prior input and asked users to start over the process using a larger dataset (particularly assessing the local explanations of the validation set) and a base model we provided. During the main session, participants had to use the system without help. The main session was video-recorded. Once participants finish their input session, we asked them to fill out an input survey, asking 2 questions for the “absolute” and “relative” valuations as follows:

- Q1: “[RQ2a, Absolute] I found understanding the **model’s vulnerable aspects** using DeepFuse to be ____.” (A 7-level Likert scale of usefulness. “7” is “extremely useful.”)
- Q2: “[RQ2a, Relative] Using DeepFuse, understanding the **model’s vulnerable aspects** was ____ than my current practice.” (A 7-level Likert scale of difficulty. “7” is “much easier.”)

5.1.3 Output Session. In this session, participants evaluated the performance change of the improved model with the test set. In particular, DeepFuse provided two pairwise comparisons between M and M_{exp} , and M_{base} and M_{exp} (see 4.2.5). After the short output session tutorial using a toy test set, participants started the main output session using the model they fine-tuned from their input session and the larger test set. Once users were finished with all the analysis and comfortable with their findings, we moved to the semi-structured exit interview. The interview had 9 question categories that were made to understand (1) their general perception about DeepFuse, such as the pros and cons they felt throughout the two sessions, (2) their perception of the specific perspectives, including (2-a) experiencing local explanation adjustment, (2-b) applying reasonability matrix in assessing the model performance, (2-c) features they used in day 1, (2-d) features they used in day 2, and (3) their suggestions for the better DeepFuse in the future. Same as S1, two researchers attended every interview. After the interview, they completed an output survey with 6 questions (see Q3 to Q8 below). Lastly, to check the usability of DeepFuse, we asked participants to fill out the System Usability Scale (SUS) survey [9] (see Appendix B).

- Q3: “[RQ2b, Absolute] I found the capability of DeepFuse regarding **improving the model performance** using my input was ____.” (A 7-level Likert scale of effectiveness. “7” is “extremely effective.”)
- Q4: “[RQ2b, Relative] I found the capability of DeepFuse regarding **improving the model performance** was ____ than my current practice.” (A 7-level Likert scale of effectiveness. “7” is “extremely effective.”)
- Q5: “[RQ1a, Absolute] Adjusting the saliency maps (as DeepFuse guided) can be effective in building future models.” (A 7-level Likert scale of agreement. “7” is “strongly agree.”)

- Q6: “[RQ1a, Relative] Adjusting the saliency maps (as DeepFuse guided) can practically change my model-building practice to a better form in the future.” (A 7-level Likert scale of agreement. “7” is “strongly agree”.)
- Q7: “[RQ1b, Absolute] On top of a model accuracy performance, using saliency maps (as DeepFuse guided) can provide an **effective measure for evaluating** my future model performance.” (A 7-level Likert scale of agreement. “7” is “strongly agree”.)
- Q8: “[RQ1b, Relative] On top of a model accuracy performance, using saliency maps (as DeepFuse guided) can practically change **the way I evaluate my future model performance** to a better form.” (A 7-level Likert scale of agreement. “7” is “strongly agree”.)

For the analysis of the exit interviews, we followed the similar process we applied in analyzing S1. The difference from S1 was the existence of the video recordings. The recordings were reviewed multiple times for transcription, code development, and analysis to synchronize with the notes. The codes and memos were developed by our two authors gradually as we intake more interviews. After the final interview, each of the authors developed the themes and shared them with each other, developing the consensus-based diagram that articulates the main insights we learned relevant to explaining the three RQs.

5.2 Results

In this section, we aggregated all survey and interview responses from the participants for the RQs we developed. S2 results suggest that (1) the workflow of the local explanation-based attention steering provided a diverse perspective in diagnosing model vulnerability, (2) the direct steering design helped the process of model revision straightforward, and (3) every participant enjoyed improved key model performance measures. Specific sub-tasks, how they are improved, and why the participants perceived they are improved are in Table 4. We believe these are not merely because of the Hawthorne and novelty effects since we have subjective evidence of performance improvement and assessment efficiency. We also organized the aspects that need improvement in Table 5, which we share in detail in the Discussion section.

The behavioral data we collected shows that all participants generated the model that outperforms (1) its model accuracy, (2) the overlap between the model’s focus and the relevant object types (IoU), and (3) the proportion of reasonable attention out of all images in a test set. The average accuracy of 12 users’ fine-tuned models (M_{exp}) was 82.95%, with an average IoU of 0.39 (“Intersection over Union” with respect to the attention ground truth of the user-defined gender-related object: “person”), and the average proportion of reasonable attention was 89.55% (see Fig. 5-A). All these

What was improved	How it was improved	Why it was improved	PIDs & (count)
Diagnosing CNNs	More diverse and rigorous perspectives in assessment	Reasonability matrix and IoU density charts provide more “diverse” and “rigorous” measures for assessing CNNs	P1, P2, P3, P5, P6, P8, P9, P10, P11, P12 (10)
	Reduced labor for model assessment	The GUI accelerates local explanation generation and vulnerability detection without repetitive coding	P2, P3, P5, P6, P7, P8, P9, P10, P12 (9)
Revising CNNs	Steering CNNs become more intuitive and straightforward	Direct adjustment using local explanation makes CNN steering easier in a more intuitive way than current practice	P2, P4, P9, P10 (4)
	Model accuracy	Shown through behavior data Shared during interview	All (12)
Outcomes	Model attention	Shown through behavior data Shared during interview	All (12)
	Model robustness	3 particularly remarked fairness 4 particularly remarked the approach can improve a model’s robustness against malicious attacks	P2, P5, P10 (3) P1, P2, P3, P4 (4)

Table 4. Study 2 user feedback regarding DeepFuse’s advantages.

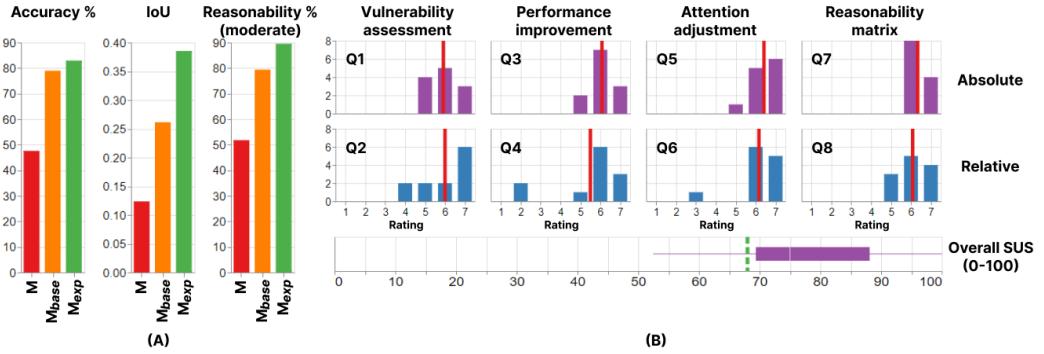


Fig. 5. Model improvement by DeepFuse and user feedback (Study 2): (A) Model performance comparison between three model conditions: M (the initial classifier), M_{base} (the state-of-the-art fine-tuned model), and M_{exp} (DeepFuse's fine-tuned model using attention). Only M_{exp} 's performance (green bars in (A)) is averaged from 12 models steered by our 12 participants. (B) User responses in two valuation perspectives (“absolute” on the first row, and “relative” on the second row) from the input survey (Q1, Q2), the output survey (Q3 to Q8), and the System Usability Scale (SUS) survey (the green dotted line in the box plot is the above-average level of 68; $Md = 75$, $M = 76.88$, $SD = 14.70$).

performances outperformed both the initial model (model M : accuracy = 47.6%, IoU = 0.12, attention reasonability = 51.8%) and the model that applied state-of-the-art fine-tuning method without attention (model M_{base} : accuracy = 79.0%, IoU = 0.26, attention reasonability = 79.4%).

Regarding the attitudinal survey data, every absolute and relative question’s mean was over 4. In terms of absolute questions, 100% of ratings were above 4-“neutral” ($M = 6.19$, $SD = 0.67$). This indicates that participants were satisfied with the overall quality of the workflow and the system. Regarding the relative questions, 89.6% of ratings were above 4-“neutral” ($M = 5.94$, $SD = 1.24$), which indicates that they felt applying the workflow and the system can practically improve their current practice.

5.2.1 [RQ1-a] Workflow: Adjusting model attention as a CNN steering method. After completing the user studies, the majority of users strongly agree that adjusting local explanations can effectively improve model performance (Q5 rating: $M = 6.42$ out of 7-“strongly agree”, $SD = 0.64$, as shown in Fig. 5-B). Also, people think their current modeling processes can be practically improved by considering the attention adjustment method (Q6 rating: $M = 6.17$ out of 7-“strongly agree”, $SD = 1.07$).

During interviews, all participants shared their positive impressions about the effectiveness of attention adjustment in improving model accuracy, which is the primary objective of conducting model fine-tuning. They also confirmed that the impact of contextual bias was reduced as attention quality increased by attention steering. By adding a new perspective from humans, a model also becomes fairer in making predictions for each target class (P2, P5, P10). Participants (P1, P2, P3, P4) with experience in model attack and defense shared the possibility of using our method to improve the robustness of the models against backdoor attacks, letting the model ignore small perturbations on an image and focus on the right area. We learned that after trying our method, people gained awareness of considering human-in-the-loop and visual-based approaches in model steering since most of the ML researchers use algorithmic approaches for handling contextual bias, such as data augmentation, hyperparameter tuning, ensemble methods, etc., rather than extensively using visualization in the fine-tuning process.

5.2.2 [RQ1-b] Workflow: Adding quality of model attention in evaluating CNNs. Based on the feedback, users agree that using an attention evaluation method (e.g., reasonability matrix as DeepFuse guided, based on Gao et al. [29]) is effective in diagnosing model vulnerabilities (Q7 rating: $M = 6.33$, $SD = 0.47$, see Fig. 5-B), and they are very likely to use this method for improving future practices Q8 rating: $M = 6.08$, $SD = 0.76$).

Participants think that the attention assessment features in DeepFuse provide more diverse and rigorous perspectives in assessing a model's vulnerabilities, especially the reasonability matrix, which can be seen as an expansion of the accuracy dimension to understanding "why" a model underperforms (P1, P3, P5, P6, P8, P9, P10, P12). P1 and P4 endorsed the necessity of equipping a reasonable matrix assessment step in checking the model's decision-making. The matrix interpretation was straightforward to most users, as it is related to the widely-used confusion matrix concept in the data science domain. The dynamic shifts of model vulnerability were well presented as shown by the reasonability matrix (3 vulnerable sub-groups, "UIA - unreasonable inaccurate", "UA - unreasonable accurate", and "RIA - reasonable inaccurate"). One major task we designed for users to achieve was the recognition of a backdoor attack in the data (i.e., added green star markers which may trigger a false prediction by the model), and all participants were able to identify the impact of the attack by evaluating attention quality using the reasonability matrix.

5.2.3 [RQ2-a] System: How DeepFuse improved CNN diagnosis. After comparing with people's current practices, DeepFuse was confirmed as a useful (Q1 rating: $M = 5.92$ out of 7- "extremely useful", $SD = 0.76$, see Fig. 5-B) and easier tool (Q2 rating: $M = 6.0$, $SD = 1.15$) in understanding model vulnerability, benefiting from the labor-efficient mechanisms.

The step-by-step nature of the assessment process in DeepFuse allows users to systematically detect both contextual and manipulated bias in the data, making it easier to reduce model vulnerability (P3, P9, P12). People believe this GUI design can significantly reduce human effort in coding and visualization management for comprehensively assessing a CNN (P2, P3, P5, P6, P7, P8, P9, P10, P12). ML engineers are well aware of the advantages of using visualization to compare metrics and surface bias, but it is a cumbersome task (e.g., repetitive file creation and loading, lack of visual-based explorers for local explanations, etc.). Instead, people mostly use command lines and unintuitive numeric comparisons for checking vulnerabilities.

One important feature that people liked was the local explanation grouping by detected objects (e.g., "person", "bicycle", etc.), which allowed them to check attention quality and accuracy changes within the common object level (P2, P3, P6, P9, P12). Some users pointed out that having consistent criteria for annotating attention quality regarding the classification task could be tricky with subjective uncertainty (P2, P4, P6, P9, P11). P6 mentioned that during the initial exploratory analysis of some models, users might not have good/bad attention criteria for annotating the attention. P10 shared an experience in exploring what objects cause contextual bias, and the biggest challenge was making a reasonable assumption at first and evaluating it over time. This challenge is critical if the annotation task is outsourced to multiple people.

5.2.4 [RQ2-b] System: How DeepFuse improved CNN revision outcomes. According to survey responses, people witnessed the highly effective capability of DeepFuse in the performance steering task (Q3 rating: $M = 6.08$ out of 7- "extremely effective", $SD = 0.64$, see Fig. 5-B). Regarding the same task, people found it slightly more effective than current approaches (Q4 rating: $M = 5.5$, $SD = 1.66$) as 2 users who preferred their approaches and rated 2- "less effective".

Aligning model attention with human perceptions can effectively revise a model performance, and with DeepFuse's adjustment mechanisms (i.e., attention drawing panel and boundary suggestions, as shown in Fig. 2), people can directly embed their intention and domain knowledge into the CNN (P2, P4, P9, P10). Regarding model performance comparison, people were able to reveal

Themes	Sub-themes	Explanations	PIDs & (count)
Existing issues	Insufficient description about reasonability matrix	It needs more description about how to interpret it for people who are unfamiliar with reasonability matrix	P4, P7, P11 (3)
	Ill-formed local explanation visualization	The local explanation visualization can be improved (especially the polygon mask)	P3, P10, P12 (3)
Future suggestions	Attention adjustment can be subjective	There could be subjective uncertainty in attention adjustment which could be avoided by providing deterministic guidelines with consistent criteria to new users	P2, P4, P6, P9, P10, P11 (6)
	Comparison on-the-fly	It would be useful if the model performance could instantly reflect with different adjustments	P7, P10 (2)
	Advanced adjustment module	The adjustment experience can be improved by supporting curve and border drawing with flexible options for adjusting weights of the masks other than binary masks	P1, P3, P6, P7 (4)
	Scalable interaction for bigger dataset	It needs improved computing capabilities and design features for bigger datasets	P2, P3, P4, P8, P11 (5)

Table 5. Study 2 user feedback regarding DeepFuse’s disadvantages.

the overall context of the image data and the corresponding impact on the model (accuracy and attention quality) by detected object sub-grouping of DeepFuse (P1, P2, P3, P5, P6, P8, P9, P11, P12). An industry practitioner who worked primarily on model quality assurance mentioned that the black-box models were not usually accessible for engineers outside the core ML team, and DeepFuse had features that could be practical for them to evaluate the model performance in that situation (P11). In the last evaluation view of DeepFuse for record-wise attention comparison (as shown on the right of Fig. 3), P7 was curious about the opposite shift of attention quality (i.e., a change from “right” to “wrong” attention after model fine-tuning) and wanted to see some quantitative measures about it.

The IoU distribution visualization was another measure in DeepFuse that could provide a rigorous comparison between model conditions (with/without attention adjustment), revealing the positive relationship between accuracy and attention quality improvement (P2, P8, P11). As people mentioned, measuring IoU was not commonly used in classification evaluation compared to segmentation tasks, and it was typically difficult to visualize.

5.3 Discussion

Overall, the system received acceptable usability [8] with an average SUS score of 76.88 ($SD = 14.70$, see the SUS box plot in Fig. 5-B, the rated scores (0-4) were converted to a 0-100 scale based on Brooke’s SUS guide [9]), exceeding the average SUS level of 68. There were 10 out of 12 participants (except P3 and P5) who gave above-average SUS scores.

Although this study is not for system-level comparison, we wanted to understand the effect of our fine-tuning mechanism collected from real users. We conducted Mann-Whitney U tests to confirm the significant performance improvement after using attention. From each of the 12 participants’ results, the accuracy of our fine-tuned model using attention was significantly greater than the baseline line condition ($U = 0$, $n_{base} = n_{exp} = 12$, $p < 0.00001$). The same results apply to the IoU and attention reasonability proportion comparisons.

Through the studies, we also identified disadvantages of our system that need to be improved (as shown in Table 5). Regarding the interpretation of the reasonability matrix produced by users’ annotation and model prediction, the guidelines can be more formally provided to be acceptable in the ML community (P4, P5, P11). The styles of attention visualization (i.e., color-scale, gray-scale, and polygon mask) need improvement, especially since the orange polygon mask was not visually clear for P3 and P10. It can be solved by having color and opacity adjustment features. People also raise the potential inconsistency issue in attention adjustment, where users may have subjective

options and criteria about where the “right” attention should be. DeepFuse needs to further provide more deterministic guidelines in attention adjustment for more complex task types, especially for tasks that require domain expertise (e.g., TB diagnosis in chest X-ray images [60]). With this uncertainty in attention adjustment, P7 and P10 suggested an instant performance comparison feature to reflect the model improvement on the fly as people annotate, which can be a future direction in active learning to have simultaneous updates while labeling in progress [90]. About the attention adjustment module, people suggested that the drawing feature should be optimized for drawing curves and near image borders, as it was not easy to do so (P1, P3, P6). P5 suggested existing smart drawing features (e.g., image matting tool in Photoshop [81]) to be added. P7 thinks that binary mask drawings might not be enough for the best attention guidance used in fine-tuning the model. A solution could be giving higher weights toward the centroid of the attention areas.

With the current data size and task setting in S2, the trade-off between manual workload and model improvement may not be as significant since the overall workload was not overwhelming and considered labor-efficient compared with existing assessment methods. Though evaluating attention maps could be a labor-intensive step, diagnosing and optimizing the model’s vulnerability were effective and easy to use based on users’ feedback. The annotation steps were incorporated with AI-supported automation (bulk annotation, object detection, object relevance filtering, adjustment recommendation, etc.) to reduce both users’ cognitive and labor workloads while gaining better performance. However, as data size increases, this labor-performance trade-off becomes essential, and more specifically, scalability solutions should be explored to reduce human labor while maintaining good fine-tuning performance. We further discussed scalability considerations regarding the trade-off in the next section (6.3).

6 IMPLICATIONS FOR DESIGN BEYOND XAI

Through S1 and S2, we learned several insights from our participants. While listening to their voice and questions, and observing the way they perceive DeepFuse after their usage, we learned that at the heart of people’s pursuit of grounding their models into their practice, one of the core challenges they encounter seems to understand how they can harmonize between the way they see the CNN should suppose to work and the way CNNs actually work. When they identify such a gap through XAI-driven tools, the upcoming challenge seemed to be to know how to reconcile such a gap efficiently and effectively. We reflect on this aspect of beyond XAI—how to help a user to shift their learned insights to actionable plans—and list up possible research directions that the HCI and CSCW communities can consider in designing future XAI or steerable AI tools to help practitioners “in the trench”.

6.1 Correlating Model Attention and Model Accuracy

One of the overarching questions we wanted to understand was how the model attention seen as reasonable by the human mind could also result in accurate prediction. Perhaps that was the reason we decided to use the reasonability matrix. If reasonable attention and accurate prediction are aligned together, the reasonable accurate instances (i.e., accurate for the right reason) and unreasonable inaccurate instances (i.e., inaccurate for the wrong reason) should increase while the unreasonable accurate and reasonable inaccurate instances should decrease. The tendency we saw was positive. We observed the reasonable accurate instances increased while the unreasonable accurate instances decreased from most participants. At least from our setting, adding more human reasoning to the model’s way of thinking has increased the model’s gaze toward intrinsic objects, resulting in an accuracy increment. However, one segment that didn’t change was the reasonable inaccurate group. We think understanding the reason when and why the model makes inaccurate predictions despite the reasonable gaze should be closely related to improving model performance.

Regarding research in Fairness, Accountability, and Transparency (FaccT), a dominant view is that human input or intervention may be required to realize a model that retains FaccT with the cost of model accuracy drop. We hope to understand the effective way to correlate the right reason, and accurate prediction can motivate the development of a fair, robust, and accurate model [42, 44]. In general, we believe it is important to understand how to align human reasoning and model accuracy. Shao et al. argue that humans “arguing” against DNNs when explanations are not reasonable can benefit the model [80]. A railroad cannot be a train [57], a snowboard is not a man [39], and a shopping cart should not be a woman [104]. Lastly, while human-guided ML has a potential and good cause [18, 30], finding a way to cut down the human-side labor is another important perspective from the two studies.

6.2 Generalizability Consideration: Beyond Binary Classification

We started to test the idea of direct steering of model attention through local explanation from the binary classification problem for reasons—simplicity of the problem and well-annotated datasets. After using DeepFuse, several participants shared their feedback and curiosity on how our pipeline can be applied in more advanced vision-based tasks. The design we provided in binary classification can be relatively simpler than the aforementioned cases. As the model’s task gets more complex and diverse, new designs customized to the particular task type and application area should be required to understand the generalizability of our findings.

Methodologically, local explanation-based attention steering is not limited to binary classification tasks. The future design can be explored to enhance CNN models for handling different tasks, such as multi-class classification, object detection, and segmentation tasks, which could possibly be expanded from processing images to videos. The core user flow beneath DeepFuse in CNN steering is as follows: First, the user flow allows human users to define reasonable and unreasonable types of attention depending on task goals. Next, the user flow motivates reasonable attention types and penalizes unreasonable attention types in a fine-tuning process suggested in Explanation-guided Learning [26]. Finally, the designer can provide a dashboard that helps users to understand how their indicated directions were reflected in the model revision process.

While the flow can be generally applicable, the way a designer facilitates a user’s definition of reasonable and unreasonable attention type should be carefully implemented depending on the type of problem. For example, in a multi-class classification or object detection task for different animals, users can employ attention logic that penalizes background and motivates foreground objects to build a more reasonable and high-performing model. As mentioned in 5.1.1, local explanation methods can be applied to different layers of a CNN to produce different levels of granularity. If the task goal requires a coarse granularity detection of a bounding box, applying local explanation visualization at the last layer of CNN can be suitable. However, if it needs more fine-grained granularity of closed curve for semantic segmentations, producing local explanations on both the first convolutional layer for edge-level of detail and the last convolutional layer for object-level detail can be considered, providing more depths of local explanation for users to evaluate. Finally, we noted P7’s suggestion about extending this flow to a more advanced video level of object classification, detection, and segmentation model steering. Due to the data volume, special design considerations need to be applied in such a task. However, upon the efficient design for indicating reasonable and unreasonable attention types, we believe that it is possible to apply the suggested flow to the problem space.

6.3 Scalability Consideration: Hundreds vs. Millions

Despite the promising performance of the model steering method, scalability remains an essential concern raised by several participants (P2, P3, P4, P8, P11), as many real-world image classification

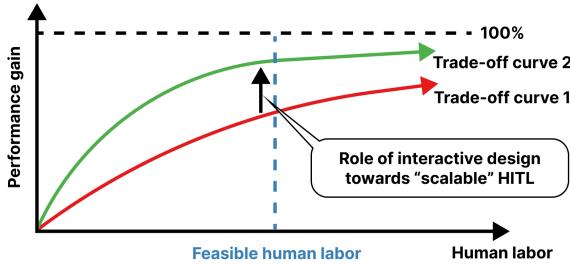


Fig. 6. An illustration of scalability optimization on a human-in-the-loop (HITL) application regarding the trade-off between human labor (as data size increases) and performance gain. The model performance can be improved faster before hitting the bottleneck of feasible human labor (the blue vertical dotted line) when the HITL processing capability is scaled up from “trade-off curve 1” (the red line) to “trade-off curve 2” (the green line).

tasks involve millions of images. Human scalability has been a crucial issue in HCI, CSCW, and beyond—while the data size can easily go up to millions and trillions in training state-of-the-art models, human cognition remains flat [97]. Even if we can surface millions of images to users, it may not be possible for them to scan images serially and achieve sensemaking. Generally, to successfully devise a scalable design, we believe that the number of images users have to go over should still not exceed thousands, and the amount of time they may spend should not exceed one hour, as recent data annotation literature suggests [14]. Herbert Simon remarked that “wealth of information creates a poverty of attention” [82].

As the trade-off between human labor and performance gain in human-in-the-loop applications is illustrated in Fig. 6, when users spend more effort as data size increases, the model will gain better performance until the workload hits the bottleneck of feasible human labor. We aim to make the curve of labor-performance trade-off steeper (from “curve 1” to “curve 2” shown in Fig. 6) through scalability optimization to improve the impact of human workload on performance gain. By devising “scalable” human-in-the-loop approaches, model performance could be further improved with the feasible amount of available human labor.

While **every** human-in-the-loop approach can suffer the bottleneck of limited information, labor source, session time, etc., ultimate breakthroughs in human-in-the-loop and interactive ML designs could come from scalability strategies. We introduce how some of the design strategies can be adopted in the design space of Beyond XAI. First, one can consider sampling from the whole dataset. Modern computer vision models can yield keywords of objects and context in the scene. Using such additional information extracted from the vast dataset, it is possible to define major and minor clusters of images. The new design may help users proceed with a small portion of sampled images derived from such clusters to reason the whole dataset and typify reasonable and unreasonable attention types accordingly. Second, one can consider examining images based on the sequence built from Active Learning, a technique that chooses the fewest unlabeled data possible that could maximize the model accuracy gain [16]. Applying active learning techniques is common in data annotation research, which can help reduce the required size of images to reason. Third, devising further intelligent features that can automate the current workflow can facilitate the process as well. Some features that need manual investigation can be automated in future designs. Finally, if there is a strong rationale for investing more human resources, one can consider crowdsourcing.

6.4 Data Iteration and Continual Lifelong Learning

DeepFuse’s capability of figuring out the vulnerability through local explanation is closely related to the capability of fortifying the dataset by adding more examples that can remove the contextual bias. Such “data iteration” is not uncommon in practice. To improve the model, the most fundamental way is to improve data. For instance, *Chameleon* lets users compare data features, training/testing splits, and performance across data versions [41]. When combining the data iteration with model steering using local explanations, one could derive some interesting design ideas that can help ML engineers to better find, search, and add the dataset.

While improving the model with new data can be straightforward, a few issues need to be considered when steering models through local explanations. First, it is necessary to understand what learning strategy can be more effective between the case where stacking every dataset in one place and retraining the model and the case of iteratively adding the new dataset and making the model “evolve”. In general, the first case can yield a high-performing model than the second case due to the chance of catastrophic forgetting, which is a problematic and almost inevitable drawback [51]. In recent years, the concept of continual lifelong learning has emerged [70] and provided a breakthrough. Understanding which strategy can yield what strengths and weaknesses in the scenario of data iteration with local explanation reasoning would be necessary.

6.5 Improving Fine-Tuning

This work is the first study that observes how ML engineers experience techniques in the Explanation-guided Learning framework in fine-tuning their model and perceiving the difference. While we saw participants satisfied with the progress they made with the RES framework, we introduced a few directions on how the RES framework can be evolved to design an improved model steering environment in the future.

One important direction is how to design a better quantitative measurement to assess the quality of the steered attention during the fine-tuning process. Simple distance-based metrics such as Mean Squared Error (MSE) or Intersection over Union (IoU) scores that are calculated purely based on the alignment of each feature can hardly comprehensively reflect the quality of the adjusted attention, as they completely ignore the correlations among visual features. One potential remedy to this issue is also to leverage fidelity-based metrics, which aim at evaluating how faithful the model’s attention is with respect to the model’s prediction. The assumption behind this is that the ‘right’ attention should contain sufficient information for the model also to make the ‘right’ prediction [20, 66, 84]; while on the other hand, removing the attention should also lead to significant negative impact for the model to make the correct prediction [20, 46, 66]. However, it is still not clear and challenging to propose a single metric that can together measure the faithfulness and the degree of alignment with the human annotation to make a more comprehensive assessment of the attention quality.

Another possible topic is how to leverage multiple annotations from different users for a single sample [16, 17]. As obtaining more than one annotation can be helpful to boost the reliability of the human boundary for attention adjustment, it poses challenges on how to align model attention with multiple ground truth boundaries. While a simple way out can be using the 50% consensus or majority vote over all the available annotations, useful information can be lost during the aggregation. Thus, new techniques are in demand to leverage each annotation effectively.

7 CONCLUSION

In this work, we examined our inquiry of how we can design a direct feedback loop between a human and a CNN through local explanations. In particular, we designed and developed the first

interactive system to help a user adjust the local explanation results regarding the gaze of CNNs. We applied our interactive design in the problem space of contextual bias for CNN engineers. With the S1, we learned ML engineers' practical challenges and desires, converting the insights to design considerations that could improve how we use local explanations in model diagnosis and steering. With DeepFuse, we conducted S2 and found how DeepFuse can provide a better workflow and experience to CNN engineers. At the same time, we also found limitations and future research directions. In particular, we boiled down and shared in Implications for Design beyond XAI within the categories of (1) correlating model attention and model accuracy, (2) generalizability consideration, (3) scalability consideration, (4) data iteration and lifelong learning, and (5) improving fine-tuning. We hope this work can benefit researchers and practitioners who seek to understand how to make XAI-driven insights actionable in steering AI.

REFERENCES

- [1] Ashraf Abdul, Christian von der Weth, Mohan Kankanhalli, and Brian Y Lim. 2020. COGAM: measuring and moderating cognitive load in machine learning model explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [2] John G Adair. 1984. The Hawthorne effect: a reconsideration of the methodological artifact. *Journal of applied psychology* 69, 2 (1984), 334.
- [3] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 337–346.
- [4] Agathe Balayn, Christoph Lofi, and Geert-Jan Houben. 2021. Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems. *The VLDB Journal* 30, 5 (2021), 739–768.
- [5] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. 2017. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 152–162.
- [6] William Blanzeisky, Barry Smyth, and Pádraig Cunningham. 2022. Algorithmic Bias and Fairness in Case-Based Reasoning. In *International Conference on Case-Based Reasoning*. Springer, 48–62.
- [7] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. 2021. Benchmarking and survey of explanation methods for black box models. *arXiv preprint arXiv:2102.13076* (2021).
- [8] John Brooke. 2013. SUS: a retrospective. *Journal of usability studies* 8, 2 (2013), 29–40.
- [9] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
- [10] Antony Brydon. 2019. Why AI Needs Human Input (And Always Will). Retrieved September 10, 2022 from <https://www.forbes.com/sites/forbestechcouncil/2019/10/30/why-ai-needs-human-input-and-always-will/>
- [11] Kelly Caine. 2016. Local standards for sample size at CHI. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 981–992.
- [12] Aditya Chattpadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.
- [13] Justin Cheng and Michael S Bernstein. 2015. Flock: Hybrid crowd-machine learning classifiers. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. 600–611.
- [14] Minsuk Choi, Cheonbok Park, Soyoung Yang, Yonggyu Kim, Jaegul Choo, and Sungsoo Ray Hong. 2019. Aila: Attentive interactive labeling assistant for document classification through attention-based deep neural networks. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–12.
- [15] Alexandra Chouldechova and Aaron Roth. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810* (2018).
- [16] Chaeyeon Chung, Jung Soo Lee, Kyungmin Park, Junsoo Lee, Jaegul Choo, and Sungsoo Ray Hong. 2021. Understanding Human-side Impact of Sequencing Images in Batch Labeling for Subjective Tasks. *Proceedings of the ACM on Human-Computer Interaction CSCW* (2021).
- [17] John Joon Young Chung, Jean Y Song, Sindhu Kutty, Sungsoo Ray Hong, Juho Kim, and Walter S Lasecki. 2019. Efficient Elicitation Approaches to Estimate Collective Crowd Answers. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–25. <https://doi.org/10.1145/3359164>
- [18] Dennis Collaris and Jarke J van Wijk. 2020. ExplainExplore: Visual exploration of machine learning explanations. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 26–35.

- [19] Jesus M Darias, Marta Caro-Martínez, Belén Díaz-Agudo, and Juan A Recio-García. 2022. Using Case-Based Reasoning for Capturing Expert Knowledge on Explanation Methods. In *International Conference on Case-Based Reasoning*. Springer, 3–17.
- [20] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. ERASER: A benchmark to evaluate rationalized NLP models. *arXiv preprint arXiv:1911.03429* (2019).
- [21] Gordon Diaper. 1990. The Hawthorne effect: A fresh examination. *Educational studies* 16, 3 (1990), 261–267.
- [22] John J Dudley and Per Ola Kristensson. 2018. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 2 (2018), 1–37.
- [23] Shannon Leigh Eggers and Char Sample. 2020. *Vulnerabilities in Artificial Intelligence and Machine Learning Applications and Data*. Technical Report. Idaho National Lab.(INL), Idaho Falls, ID (United States).
- [24] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*. 39–45.
- [25] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. 2019. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10705–10714.
- [26] Yuyang Gao, Siyi Gu, Junji Jiang, Sungsoo Ray Hong, Dazhou Yu, and Liang Zhao. 2022. Going Beyond XAI: A Systematic Survey for Explanation-Guided Learning. *arXiv preprint arXiv:2212.03954* (2022).
- [27] Yuyang Gao, Tong Sun, Rishab Bhatt, Dazhou Yu, Sungsoo Hong, and Liang Zhao. 2021. GNES: Learning to explain graph neural networks. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 131–140.
- [28] Yuyang Gao, Tong Steven Sun, Guangji Bai, Siyi Gu, Sungsoo Ray Hong, and Zhao Liang. 2022. RES: A Robust Framework for Guiding Visual Explanation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 432–442.
- [29] Yuyang Gao, Tong Steven Sun, Liang Zhao, and Sungsoo Ray Hong. 2022. Aligning eyes between humans and deep neural network through interactive attention alignment. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–28.
- [30] Yolanda Gil, James Honaker, Shikhar Gupta, Yibo Ma, Vito D’Orazio, Daniel Garjio, Shruti Gadewar, Qifan Yang, and Neda Jahanshad. 2019. Towards human-guided machine learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 614–624.
- [31] Yolanda Gil and Bart Selman. 2019. A 20-Year Community Roadmap for Artificial Intelligence Research in the US. *arXiv preprint arXiv:1908.02624* (2019).
- [32] Max Glockner, Ivan Habernal, and Iryna Gurevych. 2020. Why do you think that? exploring faithful sentence-level rationales without supervision. *arXiv preprint arXiv:2010.03384* (2020).
- [33] Nitesh Goyal, Sungsoo Ray Hong, Regan L Mandryk, Toby Jia-Jun Li, Kurt Luther, and Dakuo Wang. 2023. SHAI 2023: Workshop on Designing for Safety in Human-AI Interactions. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*. 199–201.
- [34] Ben Green and Yiling Chen. 2019. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–24.
- [35] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.
- [36] Foad Hamidi, Morgan Klaus Scheuerman, and Stacy M Branham. 2018. Gender recognition or gender reductionism? The social implications of embedded gender recognition systems. In *Proceedings of the 2018 chi conference on human factors in computing systems*. 1–13.
- [37] Gillian R Hayes. 2014. Knowing by doing: action research as an approach to HCI. In *Ways of Knowing in HCI*. Springer, 49–68.
- [38] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.
- [39] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. 2018. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 771–787.
- [40] Yusuke Hirota, Yuta Nakashima, and Noa Garcia. 2022. Gender and Racial Bias in Visual Question Answering Datasets. *arXiv preprint arXiv:2205.08148* (2022).
- [41] Fred Hohman, Kanit Wongsuphasawat, Mary Beth Kery, and Kayur Patel. 2020. Understanding and visualizing data iteration in machine learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.
- [42] Sungsoo Ray Hong, Sonia Castelo, Vito D’Orazio, Christopher Benthune, Aecio Santos, Scott Langevin, David Jonker, Enrico Bertini, and Juliana Freire. 2020. Towards Evaluating Exploratory Model Building Process with AutoML Systems. *arXiv preprint arXiv:2009.00449* (2020).
- [43] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. 2020. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. *Proceedings of the ACM on Human-Computer Interaction* 4 (2020), 1–26.

- [44] Sungsoo Ray Hong, Jorge Piazzentin Ono, Juliana Freire, and Enrico Bertini. 2019. Disseminating Machine Learning to domain experts: Understanding challenges and opportunities in supporting a model building process. In *CHI 2019 Workshop, Emerging Perspectives in Human-Centered Machine Learning*. ACM.
- [45] Scott E Hudson and Jennifer Mankoff. 2014. Concepts, values, and methods for technical human-computer interaction research. In *Ways of Knowing in HCI*. Springer, 69–93.
- [46] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. 2021. Improving deep learning interpretability by saliency guided training. *Advances in Neural Information Processing Systems* 34 (2021), 26726–26739.
- [47] Minsuk Kahng, Pierre Y Andrews, Aditya Kalro, and Duen Horng Chau. 2017. ActiVis: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 88–97.
- [48] Minsuk Kahng, Dezhong Fang, and Duen Horng Chau. 2016. Visual exploration of machine learning results using data cube analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–6.
- [49] Tharindu Kaluarachchi, Andrew Reis, and Suranga Nanayakkara. 2021. A review of recent deep learning approaches in human-centered machine learning. *Sensors* 21, 7 (2021), 2514.
- [50] Beomyoung Kim, Sangeun Han, and Junmo Kim. 2021. Discriminative region suppression for weakly-supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 1754–1761.
- [51] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.
- [52] Michael Koch, Kai von Luck, Jan Schwarzer, and Susanne Draheim. 2018. The novelty effect in large display deployments—Experiences and lessons-learned for evaluating prototypes. In *Proceedings of 16th European conference on computer-supported cooperative work-exploratory papers*. European Society for Socially Embedded Technologies (EUSSET).
- [53] Josua Krause, Aritra Dasgupta, Jordan Swartz, Yindalon Aphinyanaphongs, and Enrico Bertini. 2017. A workflow for visual diagnostics of binary classifiers using instance-level explanations. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 162–172.
- [54] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 5686–5697.
- [55] Derek Layder. 1998. *Sociological practice: Linking theory and social research*. Sage.
- [56] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jiheyeon Lee, and Jaegul Choo. 2021. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems* 34 (2021), 25123–25133.
- [57] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. 2021. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5495–5505.
- [58] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems* (2021).
- [59] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
- [60] Chang Liu, Yu Cao, Marlon Alcantara, Benyuan Liu, Maria Brunette, Jesus Peinado, and Walter Curioso. 2017. TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2314–2318.
- [61] Jiakun Liu, Qiao Huang, Xin Xia, Emad Shihab, David Lo, and Shaping Li. 2020. Is using deep learning frameworks free? characterizing technical debt in deep learning frameworks. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society*. 1–10.
- [62] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence* 2, 1 (2020), 56–67.
- [63] Yao Ming. 2017. A survey on visualization for explainable classifiers. (2017).
- [64] Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu. 2017. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 13–24.
- [65] Yao Ming, Huamin Qu, and Enrico Bertini. 2018. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 342–352.
- [66] Masahiro Mitsuhashara, Hiroshi Fukui, Yusuke Sakashita, Takanori Ogata, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. 2019. Embedding human knowledge into deep neural network via attention map. *arXiv preprint arXiv:1905.03540* (2019).

- [67] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. 2021. A multidisciplinary survey and framework for design and evaluation of explainable AI systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11, 3-4 (2021), 1–45.
- [68] Mohammed Bany Muhammad and Mohammed Yeasin. 2020. Eigen-cam: Class activation map using principal components. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.
- [69] Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.
- [70] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113 (2019), 54–71.
- [71] Haekyu Park, Nilaksh Das, Rahul Duggal, Austin P Wright, Omar Shaikh, Fred Hohman, and Duen Horng Polo Chau. 2021. Neurocartography: Scalable automatic visual summarization of concepts in deep neural networks. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 813–823.
- [72] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [73] Nicola Pezzotti. 2019. Dimensionality-Reduction Algorithms for Progressive Visual Analytics. (2019).
- [74] Donghao Ren, Saleema Amershi, Bongshin Lee, Jina Suh, and Jason D Williams. 2016. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 61–70.
- [75] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*. PMLR, 8346–8356.
- [76] Johnny Saldaña. 2015. *The coding manual for qualitative researchers*. Sage.
- [77] Aécio Santos, Sonia Castelo, Cristian Felix, Jorge Piazentin Ono, Bowen Yu, Sungsoo Ray Hong, Cláudio T Silva, Enrico Bertini, and Juliana Freire. 2019. Visus: An interactive system for automatic machine learning model building and curation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–7.
- [78] Irving Seidman. 2006. *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. Teachers college press.
- [79] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.
- [80] Xiaoting Shao, Tjitzie Rienstra, Matthias Thimm, and Kristian Kersting. 2020. Towards understanding and arguing with classifiers: Recent progress. *Datenbank-Spektrum* 20, 2 (2020), 171–180.
- [81] Gary B Shelly, Thomas J Cashman, and Joy L Starks. 2008. *Adobe Photoshop CS3: Complete concepts and techniques*. Course Technology Press.
- [82] Herbert A Simon. 1981. The sciences of the artificial, 1969. *Massachusetts Institute of Technology* (1981).
- [83] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. 2020. Don't judge an object by its context: learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11070–11078.
- [84] Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. 2021. Learning to explain: Generating stable explanations fast. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 5340–5355.
- [85] Timo Speith. 2022. A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2239–2250.
- [86] Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. 2019. explAIner: A visual analytics framework for interactive and explainable machine learning. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 1064–1074.
- [87] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. Attacking convolutional neural network using differential evolution. *IPSJ Transactions on Computer Vision and Applications* 11, 1 (2019), 1–16.
- [88] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1283–1292.
- [89] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI collaboration in data science: Exploring data scientists' perceptions of automated AI. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–24.
- [90] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. 2016. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 12 (2016), 2591–2600.
- [91] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. 2019. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *Proceedings of the IEEE/CVF International*

- Conference on Computer Vision.* 5310–5319.
- [92] Zijie J Wang, Robert Turko, and Duen Horng Chau. 2021. Dodrio: Exploring transformer models with interactive visualization. *arXiv preprint arXiv:2103.14625* (2021).
 - [93] Zijie J Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Chau. 2020. CNN 101: Interactive visual learning for convolutional neural networks. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–7.
 - [94] Greta Warren, Barry Smyth, and Mark T Keane. 2022. “Better” Counterfactuals, Ones People Can Understand: Psychologically-Plausible Case-Based Counterfactuals Using Categorical Features for Explainable AI (XAI). In *International Conference on Case-Based Reasoning*. Springer, 63–78.
 - [95] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 56–65.
 - [96] Kanit Wongsuphasawat, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mane, Doug Fritz, Dilip Krishnan, Fernanda B Viégas, and Martin Wattenberg. 2017. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 1–12.
 - [97] Chuan Yan, John Joon Young Chung, Yoon Kiheon, Yotam Gingold, Eytan Adar, and Sungsoo Ray Hong. 2022. FlatMagic: Improving Flat Colorization through AI-Driven Design for Digital Comic Professionals. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–17.
 - [98] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. 2020. DVERGE: diversifying vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems* 33 (2020), 5505–5515.
 - [99] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating how experienced UX designers effectively work with machine learning. In *Proceedings of the 2018 designing interactive systems conference*. 585–596.
 - [100] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. 2018. Grounding interactive machine learning tool design in how non-experts actually build models. In *Proceedings of the 2018 designing interactive systems conference*. 573–584.
 - [101] Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using “annotator rationales” to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*. 260–267.
 - [102] Amy X Zhang, Michael Muller, and Dakuo Wang. 2020. How do data science workers collaborate? roles, workflows, and tools. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–23.
 - [103] Zijian Zhang, Jaspreet Singh, Ujwal Gadiraju, and Avishek Anand. 2019. Dissonance between human and machine understanding. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.
 - [104] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457* (2017).

A STUDY 1 INTERVIEW QUESTIONS

About you

- Can you explain your role in your company?

Your models and development settings

- Can you explain the purpose, input, and output of your models for which you used model saliency/attention?
- Can you walk us through your process of building your model? E.g., how to collect the training set, how to train your model, how to improve your model performance, how to debug?

Use of saliency maps

- Can you explain the way you use saliency maps in understanding your model’s behavior?
- Can you explain the way you use saliency maps in supervising/improving your model’s behavior?

Working on fair/robust/accurate models

- Can you explain your experience/effort towards building more fair DNN models?
- Can you explain if attention/saliency was useful or not?

Your tools, challenge, and wish list in the future

- Can you explain the types of tools that you use for understanding/improving your DNN models?
- Can you explain the challenges you experience while interacting with your DNN?
- What new tools/features do you wish to have in the near future to make your life better?

B STUDY 2 SYSTEM USABILITY SCALE (SUS) SURVEY [9]

Indicate your degree of agreement for each of the 10 statements (on a Likert scale from 1-“strongly disagree” to 5-“strongly agree”)

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.