# Toward General Design Principles for Generative AI Applications

Justin D. Weisz[1], Michael Muller[2], Jessica He[3] and Stephanie Houde[2]

[1]IBM Research AI, Yorktown Heights, NY, USA
[2]IBM Research AI, Cambridge, MA, USA
[3]IBM Research AI, Seattle, WA, USA

Abstract
Generative AI technologies are growing in power, utility, and use. As generative technologies are being incorporated into mainstream applications, there is a need for guidance on how to design those applications to foster productive and safe use. Based on recent research on human-AI co-creation within the HCI and AI communities, we present a set of seven principles for the design of generative AI applications. These principles are grounded in an environment of *generative variability*. Six principles are focused on *designing for* characteristics of generative AI: multiple outcomes & imperfection; exploration & control; and mental models & explanations. In addition, we urge designers to design *against* potential harms that may be caused by a generative model's hazardous output, misuse, or potential for human displacement. We anticipate these principles to usefully inform design decisions made in the creation of novel human-AI applications, and we invite the community to apply, revise, and extend these principles to their own work.

**Keywords**
generative AI, design principles, human-centered AI, foundation models

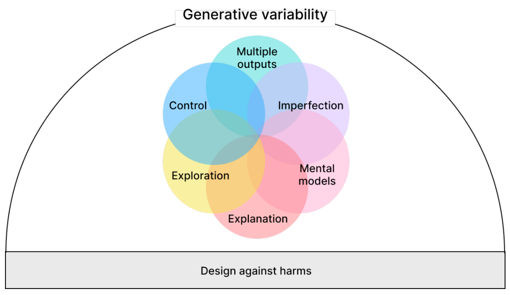

**Figure 1:** Seven principles for the design of generative AI systems. Six of these principles are presented in overlapping circles, indicating their relationships to each other. One principle stands alone, the directive to design against potential harms that may be caused by a generative model's output, misuse, or other harmful effects. These principles are bounded in an environment of *generative variability*, in which the outputs of a generative AI application may vary in quantity, quality, character, or other characteristics.

## 1. Introduction

As generative AI technologies continue to grow in power and utility, their use is becoming more mainstream. Generative models, including LLM-based foundation models [1], are being used for applications such as general Q&A (e.g. ChatGPT[1]), software engineering assistance (e.g. Copilot[2]), task automation (e.g. Adept[3]), copywriting (e.g. Jasper.ai[4]), and the creation of high-fidelity artwork (e.g. DALL-E 2 [2], Stable Diffusion [3], Midjourney[5]). Given the explosion in popularity of these new kinds of generative applications, there is a need for guidance on how to design those applications to foster productive and safe use, in line with human-centered AI values [4].

Fostering productive use is a challenge, as revealed in a recent literature survey by Campero et al. [5]. They found that many human-AI collaborative systems failed to achieve positive synergy – the notion that a human-AI team is able to accomplish superior outcomes above either party working alone. In fact, some studies have found the opposite effect, that human-AI teams produced inferior results to either a human or AI working alone [6, 7, 8, 9].

Fostering safe use is a challenge because of the potential risks and harms that stem from generative AI, either because of how the model was trained (e.g. [10])

[1]http://chat.openai.com
[2]http://copilot.github.com
[3]http://adept.ai
[4]http://jasper.ai
[5]http://midjourney.com

*Joint Proceedings of the ACM IUI Workshops 2023, March 2023, Sydney, Australia*
jweisz@us.ibm.com (J. D. Weisz); michael_muller@us.ibm.com (M. Muller); jessicahe@ibm.com (J. He); Stephanie.Houde@ibm.com (S. Houde)
0000-0003-2228-2398 (J. D. Weisz); 0000-0001-7860-163X (M. Muller); 0000-0003-2368-0099 (J. He); 0000-0002-0246-2183 (S. Houde)

or because of how it is applied (e.g. [11, 12]).

In order to address these issues, we propose a set of design principles to aid the designers of generative AI systems. These principles are grounded in an environment of **generative variability**, indicating the two properties of generative AI systems inherently different from traditional discriminative[6] AI systems: **generative**, because the aim of generative AI applications is to *produce artifacts* as outputs, rather than *determine decision boundaries* as discriminative AI systems do, and **variability**, indicating the fact that, for a given input, a generative system may produce a variety of possible outputs, many of which may be valid; in the discriminative case, it is expected that the output of a model does not vary for a given input.

We note that our principles are meant to generally apply to generative AI applications. Other sets of design principles exist for specific kinds of generative AI applications, including Liu and Chilton's guidelines for engineering prompts for text-to-image models [13], and advice about one-shot prompts for generation of texts of different kinds [14, 15, 16]. There are also more general AI-related design guidelines [17, 18, 19, 20, 21].

Six of our principles are presented as "design for..." statements, indicating the characteristics that designers should keep in mind when making important design decisions. One is presented as a "design against..." statement, directing designers to design *against* potential harms that may arise from hazardous model outputs, misuse, potential for human displacement, or other harms we have not yet considered. The principles interact with each other in complex ways, schematically represented via overlapping circles in Figure 1. For example, the characteristic denoted in one principle (e.g. multiple outputs) can sometimes be leveraged as a strategy for addressing another principle (e.g. exploration). Principles are also connected by a user's aims, such as producing a singular artifact, seeking inspiration or creative ideas, or learning about a domain. They are also connected by design features or attributes of a generative AI application, such as the support for versioning, curation, or sandbox environments.

Our aim for these principles is threefold: (1) to provide the designers of generative AI applications with the language to discuss issues unique to *generative* AI; (2) to provide strategies and guidance to help designers make important design decisions around how end users will interact with a generative AI application; and (3) to sensitize designers to the idea that generative AI applications may cause a variety of harms (likely inadvertently, but possibly intentionally). We hope these principles provide the human-AI co-creation community with a reasoned way to think through the design of novel generative AI applications.

[6]Our use of the term *discriminative* is to indicate that the task conducted by the AI algorithm is one of determining to which class or group a data instance belongs; classification and clustering algorithms are examples of discriminative AI. Although our use of the term *discriminative* may evoke imagery of human discrimination (e.g. via racial, religious, gender identity, genetic, or other lines), our use follows the scientific convention established in the machine learning community (see, e.g., https://en.wikipedia.org/wiki/Discriminative_model)

## 2. Design Principles for Generative AI Applications

We developed seven design principles for generative AI applications based on recent research in the HCI and AI communities, specifically around human-AI co-creative processes. We conducted a literature review of research studies, guidelines, and analytic frameworks from these communities [17, 18, 19, 22, 23, 20, 21, 24, 25, 26, 27, 28, 29, 30], which included experiments in human-AI co-creation [31, 32, 33, 34, 35, 36, 37], examinations of representative generative applications [38, 39, 40, 34, 41, 2, 3, 42], and a review of publications in recent workshops [43, 44, 45, 46].

### 2.1. The Environment: Generative Variability

Generative AI technologies present unique challenges for designers of AI systems compared to discriminative AI systems. First, generative AI is *generative* in nature, which means their purpose is to produce artifacts as output, rather than decisions, labels, classifications, and/or decision boundaries. These artifacts may be comprised of different types of media, such as text, images, audio, animations or videos. Second, the outputs of a generative AI model are *variable* in nature. Whereas discriminitive AI aims for deterministic outcomes, generative AI systems may not produce the same output for a given input each time. In fact, *by design*, they can produce multiple and divergent outputs for a given input, some or all of which may be satisfactory to the user. Thus, it may be difficult for users to achieve replicable results when working with a generative AI application.

Although the very nature of generative applications violates the common HCI principle that a system should respond consistently to a user's input (for critiques of this position, see [47, 48, 49, 50, 51, 12]), we take the position that this environment in which generative applications operate – *generative variability* – is a core strength. Generative applications enable users to explore or populate a "space" of possible outcomes to their query. Sometimes, this exploration is explicit, as in the case of systems that enable latent space manipulations of an artifact. Other times, exploration of a space occurs when a generative model produces multiple candidate outputs for a given input, such as multiple distinct images for a given prompt

[2, 3] or multiple implementations of a source code program [36, 37]. Recent studies also show how users may improve their knowledge of a domain by working with a generative model and its variable outputs [36, 42].

This concept of generative variability is crucially important for designers of generative AI applications to communicate to users. Users who approach a generative AI system without understanding its probabilistic nature and its capacity to produce varied outputs will struggle to interact with it in productive ways. The design principles we outline in the following sections – designing for multiple outcomes & imperfection, for exploration & human control, and for mental models & explanations – are all rooted in the notion that generative AI systems are distinct and unique because they operate in an environment of generative variability.

## 2.2. Design for Multiple Outputs

Generative AI technologies such as encoder-decoder models [52, 53], generative adversarial networks [54], and transformer models [55] are probabilistic in nature and thus are capable of producing multiple, distinct outputs for a user's input. Designers therefore need to understand the extent to which these multiple outputs should be visible to users. Do users need the ability to annotate or curate? Do they need the ability to compare or contrast? How many outputs does a user need?

Understanding the user's task can help answer these questions. If the user's task is one of *production*, in which the ultimate goal is to produce a single, satisfying artifact, then designs that help the user filter and visualize differences may be preferable. For example, a software engineer's goal is often to implement a method that performs a specific behavior. Tools such as Copilot take a user's input, such as a method signature or documentation, and provide a singular output. Contrarily, if the user's task is one of *exploration*, then designs that help the user curate, annotate, and mutate may be preferable. For example, a software engineer may wish to explore a space of possible test cases for a code module. Or, an artist may wish to explore different compositions or styles to see a broad range of possibilities. Below we discuss a set of strategies for helping design for multiple outputs.

### 2.2.1. Versioning

Because of the randomness involved in the generative process, as well as other user-configurable parameters (e.g. a random seed, a temperature, or other types of user controls), it may be difficult for a user to produce exactly the same outcome twice. As a user interacts with a generative AI application and creates a set of outputs, they may find that they prefer earlier outputs to later ones. How can they recover or reset the state of the system to generate such earlier outputs? One strategy is to keep track of all of these outputs, as well as the parameters that produced them, by versioning them. Such versioning can happen manually (e.g. the user clicks a button to "save" their current working state) or automatically.

### 2.2.2. Curation

When a generative model is capable of producing multiple outputs, users may need tools to curate those outputs. Curation may include collecting, filtering, sorting, selecting, or organizing outputs (possibly from the versioned queue) into meaningful subsets or groups, or creating prioritized lists or hierarchies of outputs according to some subjective or objective criteria. For example, CogMol[7] generates novel molecular compounds, which can be sorted by various properties, such as their molecular weight, toxicity, or water solubility [56, 57]. In addition, the confidence of the model in each output it produced may be a useful way to sort or rank outputs, although in some cases, model confidence scores may not be indicative of the quality of the model's output [32].

### 2.2.3. Annotation

When a generative model has produced a large number of outputs, users may desire to add marks, decorators, or annotations to outputs of interest. These annotations may be applied to the output itself (e.g. "I like this") or it may be applied to a portion or subset of the output (e.g. flagging lines of source code that look problematic and need review).

### 2.2.4. Visualizing Differences

In some cases, a generative model may produce a diverse set of distinct outputs, such as images of cats that look strikingly different from each other. In other cases, a generative model may produce a set of outputs for which it is difficult to discern differences, such as a source code translation from one language to another. In this case, tools that aid users in visualizing the similarities and differences between multiple outputs can be useful. Depending on the users' goals, they may seek to find the *invariant* aspects across outcomes, such as identifying which parts of a source code translation were the same across multiple translations, indicating a confidence in its correctness. Or, users may prioritize the *variant* aspects for greater creativity and inspiration. For example, Sentient Sketchbook [58] is a video game co-creation system that displays a number of different metrics of the maps it generates, enabling users to compare newly-generated maps with their current map to understand how they differ.

[7]http://covid19-mol.mybluemix.net

## 2.3. Design for Imperfection

It is highly important for users to understand that the quality of a generative model's outputs will vary. Users who expect a generative AI application to produce exactly the artifact they desire will experience frustration when they work with the system and find that it often produces imperfect artifacts. By "imperfect," we mean that the artifact itself may have imperfections, such as visual misrepresentations in an image, bugs or errors in source code, missing desired elements (e.g. "an illustration of a bunny with a carrot" fails to include a carrot), violations of constraints specified in the input prompt (e.g. "write a 10 word sentence" produces a much longer or shorter sentence), or even untruthful or misleading answers (e.g. a summary of a scientific topic that includes non-existent references [59]). But, "imperfect" can also mean "doesn't satisfy the user's desire," such as when the user prompts a model and doesn't get back any satisfying outputs (e.g. the user didn't like any of the illustrations of a bunny with a carrot). Below we discuss a set of strategies for helping design for imperfection.

### 2.3.1. Multiple Outputs

Our previous design principle is also a strategy for handling imperfect outputs. If a generative model is allowed to produce multiple outputs, the likelihood that one of those outputs is satisfying to the user is increased. One example of this effect is in how code translation models are evaluated, via a metric called $pass@k$ [60, 61]. The idea is that the model is allowed to produce $k$ code translations for a given input, and if any of them pass a set of unit tests, then the model is said to have produced a correct translation. In this way, generating multiple outputs serves to mitigate the fact that the model's most-likely output may be imperfect. However, it is left up to the user to review the set of outputs and identify the one that is satisfactory; with multiple outputs that are very similar to each other, this task may be difficult [37], implying the need for a way to easily visualize differences.

### 2.3.2. Evaluation & Identification

Given that generative models may not produce perfect (or perfectly satisfying) outputs, they may still be able to provide users with a signal about the quality of its output, or indicate parts that require human review. As previously discussed, a model's per-output confidence scores may be used (with care) to indicate the quality of a model's output. Or, domain-specific metrics (e.g. molecular toxicity, compiler errors) may be useful indicators to evaluate whether an artifact achieved a desirable level of quality. Thus, evaluating the quality of generated artifacts and identifying which portions of those artifacts may contain imperfections (and thus require human review, discussed further in Weisz et al. [36]) can be an effective way for handling imperfection.

### 2.3.3. Co-Creation

User experiences that allow for co-creation, in which both the user and the AI can edit a candidate artifact, will be more effective than user experiences that assume or aim for the generative model to produce a perfect output. Allowing users to edit a model's outputs provides them with the opportunity to find and fix imperfections, and ultimately achieve a satisfactory artifact. One example of this idea is Github Copilot [62], which is embedded in the VSCode IDE. In the case when Copilot produces an imperfect block of source code, developers are able to edit it *right in context* without any friction. By contrast, tools like Midjourney or Stable Diffusion only produce a gallery of images to chose from; editing those images requires the user to shift to a different environment (e.g. Photoshop).

### 2.3.4. Sandbox / Playground Environment

A sandbox or playground environment ensures that when a user interacts with a generated artifact, their interactions (such as edits, manipulations, or annotations) do not impact the larger context or environment in which they are working. Returning to the example of Github Copilot, since it is situated inside a developer's IDE, code it produces is directly inserted into the working code file. Although this design choice enables co-creation, it also poses a risk that imperfect code is injected into a production code base. A sandbox environment that requires users to explicitly copy and paste code in order to commit it to the current working file may guard against the accidental inclusion of imperfect outputs in a larger environment or product.

## 2.4. Design for Human Control

Keeping humans in control of AI systems is a core tenet of human-centered AI [63, 64, 4]. Providing users with controls in generative applications can improve their experience by increasing their efficiency, comprehension, and ownership of generated outcomes [34]. But, in co-creative contexts, there are multiple ways to interpret what kinds of "control" people need. We identify three kinds of controls applicable to generative AI applications.

### 2.4.1. Generic Controls

One aspect of control relates to the exploration of a design space or range of possible outcomes (as discussed in Section 2.5). Users need appropriate controls in order to drive their explorations, such as control over the number of outputs produced from an input or the amount of variability present in the outputs. We refer to these kinds of controls as *generic controls*, as they are applicable to any particular generative technology or domain. As an example, some generative projects may involve a "lifecycle" pattern in which users benefit from seeing a great diversity of outputs in early stages of the process in order to search for ideas, inspirations, or directions. Later stages of the project may focus on a smaller number (or singular) output, requiring controls that specifically operate on that output. Many generative algorithms include a user-controllable parameter called *temperature*. A low temperature setting produces outcomes that are very similar to each other; conversely, a high temperature setting produces outcomes that are very dissimilar to each other. In the "lifecycle" model, users may first set a high temperature for increased diversity, and then reduce it when they wish to focus on a particular area of interest in the output space. This effect was observed in a study of a music co-creation tool, in which novice users dragged temperature control sliders to the extreme ends to explore the limits of what the AI could generate [34].

#### 2.4.2. Technology-specific Controls

Other types of controls will depend on the particular generative technology being employed. Encoder-decoder models, for example, often allow users to perform latent space manipulations of an artifact in order to control semantically-meaningful attributes. For example, Liu and Chilton [65] demonstrate how semantic sliders can be used to control attributes of 3D models of animals, such as the animal's torso length, neck length, and neck rotation. Transformer models use a temperature parameter to control the amount of randomness in the generation process [66]. Natural language prompting, and the emerging discipline of prompt engineering [13], provide additional ways to tune or tweak the outputs of large language models. We refer to these kinds of controls as *technology-specific controls*, as the controls exposed to a user in a user interface will depend upon the particular generative AI technology used in the application.

#### 2.4.3. Domain-specific Controls

Some types of user controls will be domain-specific, dependent on the type of artifact being produced. For example, generative models that produce molecules as output might be controlled by having the user specify desired properties such as molecular weight or water solubility; these types of constraints might be propagated to the model itself (e.g. expressed as a constraint in the encoder phase), or they may simply act as a filter on the model's output (e.g. hide anything from the user that doesn't satisfy the constraints). In either case, the control itself is dependent on the fact that the model is producing a specific kind of artifact, such as a molecule, and would not logically make sense for other kinds of artifacts in other domains (e.g. how would you control the water solubility for a text-to-image model?). Thus, we refer to these types of controls, independent of how they are implemented, as *domain specific*. Other examples of domain-specific controls include the reading level of a text, the color palette or artistic style of an image, or the run time or memory efficiency of source code.

## 2.5. Design for Exploration

Because users are working in an environment of generative variability, they will need some way to "explore" or "navigate" the space of potential outputs in order to identify one (or more) that satisfies their needs. Below we discuss a set of strategies for helping design for exploration.

#### 2.5.1. Multiple Outputs

The ability for a generative model to produce multiple outputs (Section 2.2) is an enabler of exploration. Returning to the bunny and carrot example, an artist may wish to explore different illustrative styles and prompt (and re-prompt) the model for additional candidates of "a bunny with a carrot" in various kinds of styles or configurations. Or, a developer can explore different ways to implement an algorithm by prompting (and re-prompting) a model to produce implementations that possess different attributes (e.g. "implement this using recursion," "implement this using iteration," or "implement this using memoization"). In this way, a user can get a sense of the different possibilities the model is capable of producing.

#### 2.5.2. Control

Depending on the specific technical architecture used by the generative application, there may be different ways for users to control it (Section 2.4). No matter the specific mechanisms of control, *providing controls* to a user provides them with the ability to interactively work with the model to explore the space of possible outputs for their given input.

#### 2.5.3. Sandbox / Playground Environment

A sandbox or playground environment can enable exploration by providing a separate place in which new candidates can be explored, without interfering with a user's main working environment. For example, in a project using Copilot, Cheng et al. [67] suggest providing, "a sandbox mechanism to allow users to play with the prompt in the context of their own project."

#### 2.5.4. Visualization

One way to help users understand the space in which they are exploring is to explicitly visualize it for them. Kreminski et al. [33] introduce the idea of expressive range coverage analysis (ERCA) in which a user is shown a visualization of the "range" of possible generated artifacts across a variety of metrics. Then, as users interact with the system and produce specific artifact instances, those instances are included in the visualization to show how much of the "range" or "space" was explored by the user.

### 2.6. Design for Mental Models

Users form mental models when they work with technological systems [68, 69, 70]. These models represent the user's understanding of how the system works and how to work with it effectively to produce the outcomes they desire. Due to the environment of generative variability, generative AI applications will pose new challenges to users because these applications may violate existing mental models of how computing systems behave (i.e. in a deterministic fashion). Therefore, we recommend designing to support users in creating accurate mental models of generative AI applications in the following ways.

#### 2.6.1. Orientation to Generative Variability

Users may need a general introduction to the concept of generative AI. They should understand that the system may produce multiple outputs for their query (Section 2.2), that those outputs may contain flaws or imperfections (Section 2.3), and that their effort may be required to collaborate *with* the system in order to produce desired artifacts via various kinds of controls (Section 2.4).

#### 2.6.2. Role of the AI

Research in human-AI interaction suggests that users may view an AI application as filling a role such as an assistant, coach, or teammate [29]. In a study of video game co-creation, Guzdial et al. [71] found participants to ascribe roles of friend, collaborator, student, and manager to the AI system. Recent work by Ross et al. [42] examined software engineers' role orientations toward a programming assistant and found that people viewed the assistant with a tool orientation, but interacted with it as if it were a social agent. Clearly establishing the role of a generative AI application in a user's workflow, as well as its level of autonomy (e.g. [72, 73, 74, 75]), will help users better understand how to interact effectively with it. Designers can reason about the role of their application by answering questions such as, is it a tool or partner? does it act proactively or does it just respond to the user? does it make changes to an artifact directly or does it simply make recommendations for the user?

### 2.7. Design for Explanations

Generative AI applications will be unfamiliar and possibly unusual to many users. They will want to know what the application can (and cannot) do, how well it works, and how to work with it effectively. Some users may even wish to understand the technical details of how the underlying generative AI algorithms work, although these details may not be necessary to work effectively with the model (as discussed in [36]).

In recent years, the explainable AI (XAI) community has made tremendous progress at developing techniques for explaining how AI systems work [76, 77, 21, 78, 79]. Much of the work in XAI has focused on discriminative algorithms: how they generally make decisions (e.g. via interpretable models [80, Chapter 5] or feature importance [80, Section 8.5], and why they make a decision in a specific instance (e.g. via counterfactual explanations [80, Section 9.3].

Recent work in human-centered XAI (HCXAI) has emphasized designing explanations that cater to human knowledge and human needs [77]. This work grew out of a general shift toward human-centered data science [47], in which the import of explanations is not for a technical user (data scientist), but for an end user who might be impacted by a machine learning model.

In the case of generative AI, recent work has begun to explore the needs for explainability. Sun et al. [35] explored explainability needs of software engineers working with a generative AI model for various types of use cases, such as code translation and autocompletion. They identified a number of types of questions that software engineers had about the generative AI, its capabilities, and its limitations, indicating that explainability is an important feature for generative AI applications. They also identified several gaps in existing explainability frameworks stemming from the *generative* nature of the AI system, indicating that existing XAI techniques may not be sufficient for generative AI applications. Thus, we make the following recommendations for how to design for explanations.

#### 2.7.1. Calibrate Trust by Communicating Capabilities and Limitations

Because of the inherent imperfection of generative AI outputs, users would be well-served if they understood the limitations of these systems [81, 82], allowing them to *calibrate* their trust in terms of what the application can and cannot do [83]. When these kinds of imperfections (Section 2.3) are not signaled, users of co-creative tools may mistakenly blame themselves for shortcomings of generated artifacts in co-creative applications [34], and users in Q & A use cases can be shown deceptive misconceptions and harmful falsehoods as objective answers [84]. One way to communicate the capabilities of a generative AI application is to show examples of what it can do. For example, Midjourney provides a public discussion space to orient new users and show them what other users have produced with the model. This space not only shows the outputs of the model (e.g. images), but the textual prompts that produced the images. In this way, users can more quickly come to understand how different prompts influence the application's output. To communicate limitations, systems like ChatGPT contain modal screens to inform users of the system's limitations.

#### 2.7.2. Use Explanations to Create and Reinforce Accurate Mental Models

Weisz et al. [36] explored how a generative model's confidence could be surfaced in a user interface. Working with a transformer model on a code translation task, they developed a prototype UI that highlighted tokens in the translation that the model was not confident in. In their user study, they found that those highlights also served as explanations for how the model worked: users came to understand that each source code token was chosen probabilistically, and that the model had considered other alternatives. This design transformed an algorithmic weakness (imperfect output) into a resource for users to understand how the algorithm worked, and ultimately, to control its output (by showing users where they might need to make changes).

### 2.8. Design Against Harms

The use of AI systems – including generative AI applications – may unfortunately lead to diverse forms of harms, especially for people in vulnerable situations. Much work in AI ethics communities has identified how discriminative AI systems may perpetuate harms such as the denial of personhood or identity [49, 85, 86]; the deprivation of liberty or children [87, 88], and the erasure of persons, cultures, or nations through data silences [81]. We identify four types of potential harms, some of which are unique to the generative domain, and others which represent existing risks of AI applications that may manifest in new ways.

Our aim in this section is to sensitize designers to the potential risks and harms that generative AI systems may pose. We do not prescribe solutions to address these risks, in part because it is an active area of research to understand how these kinds of risks could be mitigated. Risk identification, assessment, and mitigation is a sociotechnical problem involving computing resources, humans, and cultures. Even with our focus on the design of generative applications, an analysis of harms that is limited to design concepts may blur into technosolutionism [89, 90, 91].

We do posit that *human-centered* approaches to generative AI design are a useful first step, but must be part of a larger strategy to understand who are the direct and indirect stakeholders of a generative application [92, 93], and to work directly with those stakeholders to *identify* harms, *understand* what are their differing priorities and value tensions [94], and *negotiate* issues of culture, policy, and (yes) technology to meet these diverse challenges (e.g., [95, 96, 97]).

#### 2.8.1. Hazardous Model Outputs

Generative AI applications may produce artifacts that cause harm. In an integrative survey paper, Weidinger et al. [10] list six types of potential harms of large language models, three of which regard the harms that may be caused by the model's output:

- **Discrimination, Exclusion, and Toxicity**. Generative models may produce outputs that promote discrimination against certain groups, exclude certain groups from representation, or produce toxic content. Examples include text-to-image models that fail to produce ethnically diverse outputs for a given input (e.g. a request for images of doctors produces images of male, white doctors [98] or language models that produce inappropriate language such as swear words, hate speech, or offensive content [17, 20].
- **Information Hazards**. Generative models may inadvertently leak private or sensitive information from their training data. For example, Carlini et al. [99] found that strategically prompting GPT-2 revealed an individual's full name, work address, phone number, email, and fax number. Additionally, larger models may be more vulnerable to these types of attacks [99, 100].
- **Misinformation Harms**. Generative models may produce inaccurate misinformation in response to a user's query. Lin et al. [84] found that GPT-3 can provide false answers that mimic human falsehoods and misconceptions, such as "coughing can help stop a heart attack" or "[cold weather] tells us that global warming is a hoax". Singhal et al. [101] caution against the tendency of LLMs to hallucinate references, especially if consulted for medical decisions. Albrecht et al. [102] claim that LLMs have few defenses against adversarial attacks while advising about ethical questions. The Galactica model was found to hallucinate non-existent scientific references [103], and Stack Overflow has banned responses sourced from ChatGPT due to their high rate of incorrect, yet plausible, responses [104].

In addition to those harms, a generative model's outputs may be hazardous in other ways as well.

- **Deceit, Impersonation, and Manipulation**. Generative algorithms can be used to create false records or "deep fakes" (e.g., [11, 105]), to impersonate others (e.g. [106]), or to distort information into politically-altered content [107]. In addition, they may manipulate users who believe that they are chatting with another human rather than with an algorithm, as in the case of an unreviewed ChatGPT "experiment" in which at least 4,000 people seeking mental health support were connected to a chatbot rather than a human counselor [108].
- **Copyright, Licenses, and Intellectual Property**. Generative models may have been trained on data protected by regulations such as the GDPR, which prohibits the re-use of data beyond the purposes for which it was collected. In addition, large language models have been referred to as "stochastic parrots" due to their ability to reproduce data that was used during their training [109]. One consequence of this effect is that the model may produce outputs that incorporate or remix materials that are subject to copyright or intellectual property protections [110, 111, 112]. For example, the Codex model, which produces source code as output, may (re-)produce source code that is copyrighted or subject to a software license, or that was openly shared under a creative commons license that prohibits commercial re-use (e.g., in a pay-to-access LLM). Thus, the use of a model's outputs in a project may cause that project to violate copyright protections, or subject that project to a restrictive license (e.g. GPL). As of this writing, there is a lawsuit against GitHub, Microsoft, and OpenAI on alleged copyright violations in the training of Codex [113].

#### 2.8.2. Misuse

Weidinger et al. [10] describe how generative AI applications may be misused in ways unanticipated by the creators of those systems. Examples include making disinformation cheaper and more effective, facilitating fraud and scams, assisting code generation for cyberattacks, or conducting illegitimate surveillance and censorship. In addition to these misuses, Houde et al. [11] also identify business misuses of generative AI applications such as facilitating insurance fraud and fabricating evidence of a crime. Although designers may not be able to prevent users from intentionally misusing their generative AI applications, there may be preventative measures that make sense for a given application domain. For example, output images may be watermarked to indicate they were generated by a particular model, blocklists may be used to disallow undesirable words in a textual prompt, or multiple people may be required to review or approve a model's outputs before they can be used.

#### 2.8.3. Human Displacement

One consequence of the large-scale deployment of generative AI technologies is that they may come to *replace*, rather than *augment* human workers. Such concerns have been raised in related areas, such as the use of automated AI technologies in data science Wang et al. [114, 115]. Weidinger et al. [10] specifically discuss the potential economic harms and inequalities that may arise as a consequence of widespread adoption of generative AI. If a generative model is capable of producing high-fidelity outputs that rival (or even surpass) what can be created by human effort, are the humans necessary anymore? Contemporary fears of human displacement by generative technologies are beginning to manifest in mainstream media, such as in the case of illustrators' concerns that text-to-image models such as Stable Diffusion and Midjourney will put them out of a job [116]. We urge designers to find ways to design generative AI applications that *enhance* or *augment* human abilities, rather than applications that aim to *replace* human workers. Copilot serves as one example of a tool that clearly enhances the abilities of a software engineer: it operates on the low-level details of a source code implementation, freeing up software engineers to focus more of their attention on higher-level architectural and system design issues.

## 3. Discussion

### 3.1. Designing for User Aims

Users of generative AI applications may have varied aims or goals in using those systems. Some users may be in pursuit of *perfecting a singular artifact*, such as a method implementation in a software program. Other users may be in *pursuit of inspiration or creative ideas*, such as when exploring a visual design space. As a consequence of working with a generative AI application, users may also *enhance their own learning or understanding* of the domain in which they are operating, such as when a software engineer learns something new about a programming language from the model's output. Each of these aims can be supported by our design principles, as well as help designers determine the appropriate strategy for addressing the challenges posed by each principle.

To support artifact production, designers ought to carefully consider how to manage a model's multiple, imperfect outputs. Interfaces ought to support users in curating, annotating, and mutating artifacts to help users refine a singular artifact. The ability to version artifacts, or show a history of artifact edits, may also be useful to enable users to revisit discarded options or undo undesirable modifications. For cases in which users seek to produce one "ideal" artifact that satisfies some criteria, controls that enable them to co-create with the generative tool can help them achieve their goal more efficiently, and explanations that signal or identify imperfections can tell them how close or far they are from the mark.

To support inspiration and creativity, designers also ought to provide adequate controls that enable users to explore a design space of possibilities [33, 117]. Visualizations that represent the design space can also be helpful as they can show which parts the user has vs. has not explored, enabling them to explore the novel parts of that space. Tools that help users manage, curate, and filter the different outputs created during their explorations can be extremely helpful, such as a digital mood board for capturing inspiring model outputs.

Finally, to support learning how to effectively interact with a generative AI application, designers ought to help users create accurate mental models [118] through explanations [76, 77, 21, 78, 79]. Explanations can help answer general questions such as what a generative AI application is capable or not capable of generating, how the model's controls impact its output, and how the model was trained and the provenance of its training data. They can also answer questions about a specific model output, such as how confident the model was in that output, which portions of that output might need human review or revision, how to adjust or modify the input or prompt to adjust properties of the output, or what other options or alternatives exist for that output.

### 3.2. The Importance of Value-Sensitive Design in Mitigating Potential Harms

Designers need to be sensitive to the potential harms that may be caused by the rapid maturation and widespread adoption of generative AI technologies. Although sociotechnical means for mitigating these harms have yet to be developed, we recommend that designers use a Value Sensitive Design approach [92, 93] when reasoning about how to design generative AI applications. By clearly identifying the different stakeholders and impacted parties of a generative AI application, and explicitly enumerating their values, designers can make more reasoned judgments about how those stakeholders might be impacted by hazardous model outputs, model misuse, and issues of human displacement.

## 4. Limitations and Future Work

Generative AI applications are still in their infancy, and new kinds of co-creative user experiences are emerging at a rapid pace. Thus, we consider these principles to be in their infancy as well, and it is possible that other important design principles, strategies, and/or user aims have been overlooked. In addition, although these principles can provide helpful guidance to designers in making specific design decisions, they need to be validated in real-world settings to ensure their clarity and utility.

## 5. Conclusion

We present a set of seven design principles for generative AI applications. These principles are grounded in an environment of generative variability, the key characteristics of which are that a generative AI application will generate artifacts as outputs, and those outputs may be varied in nature (e.g. of varied quality or character). The principles focus on designing for multiple outputs and the imperfection of those outputs, designing for exploration of a space or range of possible outputs and maintaining human control over that exploration, and designing to establish accurate mental models of the generative AI application via explanations. We also urge designers to design *against* the potential harms that may be caused by hazardous model output (e.g. the production of inappropriate language or imagery, the reinforcement of existing stereotypes, or a failure to inclusively represent different groups), by misuse of the model (e.g. by creating disinformation or fabricating evidence), or by displacing human workers (e.g. by designing for the *replacement* rather than the *augmentation* of human workers). We envision these principles to help designers make reasoned choices as they create novel generative AI applications.

## References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258 (2021).

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, Hierarchical text-conditional image generation with clip latents, arXiv preprint arXiv:2204.06125 (2022).

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[4] B. Shneiderman, Human-Centered AI, Oxford University Press, 2022.

[5] A. Campero, M. Vaccaro, J. Song, H. Wen, A. Almaatouq, T. W. Malone, A test for evaluating performance in human-computer systems, arXiv preprint arXiv:2206.12390 (2022).

[6] E. Clark, A. S. Ross, C. Tan, Y. Ji, N. A. Smith, Creative writing with a machine in the loop: Case studies on slogans and stories, in: 23rd International Conference on Intelligent User Interfaces, 2018, pp. 329–340.

[7] Z. Buçinca, M. B. Malaya, K. Z. Gajos, To trust or to think: cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making, Proceedings of the ACM on Human-Computer Interaction 5 (2021) 1–21.

[8] M. Jacobs, M. F. Pradier, T. H. McCoy, R. H. Perlis, F. Doshi-Velez, K. Z. Gajos, How machine-learning recommendations influence clinician treatment selections: the example of antidepressant selection, Translational psychiatry 11 (2021) 1–9.

[9] B. Kleinberg, B. Verschuere, How humans impair automated deception detection performance, Acta Psychologica 213 (2021) 103250.

[10] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, et al., Ethical and social risks of harm from language models, arXiv preprint arXiv:2112.04359 (2021).

[11] S. Houde, V. Liao, J. Martino, M. Muller, D. Piorkowski, J. Richards, J. D. Weisz, Y. Zhang, Business (mis)use cases of generative ai, in: Joint Proceedings of the Workshops on Human-AI Co-Creation with Generative Models and User-Aware Conversational Agents co-located with 25th International Conference on Intelligent User Interfaces (IUI 2020), 2020.

[12] M. Muller, S. I. Ross, S. Houde, M. Agarwal, F. Martinez, J. T. Richards, K. Talamadupula, J. D. Weisz, Drinking chai with your (AI) programming partner: A design fiction about generative AI for software engineering 107-122, in: A. Smith-Renner, O. Amir (Eds.), Joint Proceedings of the IUI 2022 Workshops: APEx-UI, HAI-GEN, HEALTHI, HUMANIZE, TExSS, SOCIALIZE co-located with the ACM International Conference on Intelligent User Interfaces (IUI 2022), Virtual Event, Helsinki, Finland, March 21-22, 2022, volume 3124 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 107–122.

[13] V. Liu, L. B. Chilton, Design guidelines for prompt engineering text-to-image generative models, in: CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–23.

[14] C. Greyling, Prompt engineering, text generation and large language models, 2022. URL: https://cobusgreyling.medium.com/prompt-engineering-text-generation-large-language-models-3d90c527c6d5.

[15] L. Reynolds, K. McDonell, Prompt programming for large language models: Beyond the few-shot paradigm, in: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–7.

[16] P. Denny, V. Kumar, N. Giacaman, Conversing with copilot: Exploring prompt engineering for solving cs1 problems using natural language, 2022. URL: https://arxiv.org/abs/2210.15157.

[17] ACM, Words matter: Alternatives for charged terminology in the computing profession, 2023. URL: https://www.acm.org/diversity-inclusion/words-matter.

[18] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al., Guidelines for human-ai interaction, in: Proceedings of the 2019 chi conference on human factors in computing systems, 2019, pp. 1–13.

[19] A. Computer, Human interface guidelines, 2022. URL: https://developer.apple.com/design/human-interface-guidelines/guidelines/overview.

[20] IBM, Racial equity in design, 2023. URL: https://www.ibm.com/design/racial-equity-in-design/.

[21] Q. V. Liao, D. Gruen, S. Miller, Questioning the ai: informing design practices for explainable ai user experiences, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–15.

[22] S. Deterding, J. Hook, R. Fiebrink, M. Gillies, J. Gow, M. Akten, G. Smith, A. Liapis, K. Compton, Mixed-initiative creative interfaces, in: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 2017, pp. 628–635.

[23] I. Grabe, M. González-Duque, S. Risi, J. Zhu, Towards a framework for human-ai interaction patterns in co-creative gan applications, Joint Proceedings of the ACM IUI Workshops 2022, March 2022, Helsinki, Finland (2022).

[24] M. L. Maher, Computational and collective creativity: Who's being creative?, in: ICCC, Citeseer, 2012, pp. 67–71.

[25] M. L. Maher, B. Magerko, D. Venura, D. Fisher, R. Cardona-rivera, N. Fulda, J. Gooth, M. Lee, D. Wilson, J. Kaufman, et al., A research plan for integrating generative and cognitive ai for human centered, explainable co-creative ai, in: ACM CHI Conference on Human Factors in Computing Systems, 2022.

[26] M. Muller, J. D. Weisz, W. Geyer, Mixed initiative generative ai interfaces: An analytic framework for generative ai applications, ICCC 2020 Workshop, The Future of Co-Creative Systems, 2020. URL: https://computationalcreativity.net/workshops/cocreative-iccc20/papers/Future_of_co-creative_systems_185.pdf.

[27] M. Muller, J. Weisz, Extending a human-ai collaboration framework with dynamism and sociality, in: 2022 Symposium on Human-Computer Interaction for Work, 2022, pp. 1–12.

[28] T. Lubart, How can computers be partners in the creative process: classification and commentary on the special issue, International Journal of Human-Computer Studies 63 (2005) 365–369.

[29] I. Seeber, E. Bittner, R. O. Briggs, T. De Vreede, G.-J. De Vreede, A. Elkins, R. Maier, A. B. Merz, S. Oeste-Reiß, N. Randrup, et al., Machines as teammates: A research agenda on ai in team collaboration, Information & management 57 (2020) 103174.

[30] A. Spoto, N. Oleynik, Library of mixed-initiative creative interfaces, 2017. URL: http://mici.codingconduct.cc/.

[31] M. Agarwal, J. J. Barroso, T. Chakraborti, E. M. Dow, K. Fadnis, B. Godoy, M. Pallan, K. Talamadupula, Project clai: Instrumenting the command line as a new environment for ai agents, arXiv preprint arXiv:2002.00762 (2020).

[32] M. Agarwal, K. Talamadupula, S. Houde, F. Martinez, M. Muller, J. Richards, S. Ross, J. D. Weisz, Quality estimation & interpretability for code translation, in: Proceedings of the NeurIPS 2020 Workshop on Computer-Assisted Programming (NeurIPS 2020), 2020.

[33] M. Kreminski, I. Karth, M. Mateas, N. Wardrip-Fruin, Evaluating mixed-initiative creative interfaces via expressive range coverage analysis., in: IUI Workshops, 2022, pp. 34–45.

[34] R. Louie, A. Coenen, C. Z. Huang, M. Terry, C. J. Cai, Novice-ai music co-creation via ai-steering tools for deep generative models, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–13.

[35] J. Sun, Q. V. Liao, M. Muller, M. Agarwal, S. Houde, K. Talamadupula, J. D. Weisz, Investigating explainability of generative ai for code through scenario-based design, in: 27th International Conference on Intelligent User Interfaces, 2022, pp. 212–228.

[36] J. D. Weisz, M. Muller, S. Houde, J. Richards, S. I. Ross, F. Martinez, M. Agarwal, K. Talamadupula, Perfection not required? human-ai partnerships in code translation, in: 26th International Conference on Intelligent User Interfaces, 2021, pp. 402–412.

[37] J. D. Weisz, M. Muller, S. I. Ross, F. Martinez, S. Houde, M. Agarwal, K. Talamadupula, J. T. Richards, Better together? an evaluation of ai-supported code translation, in: 27th International Conference on Intelligent User Interfaces, 2022, pp. 369–391.

[38] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[39] T. E. Johnson, Y. Lee, M. Lee, D. L. O'Connor, M. K. Khalil, X. Huang, Measuring sharedness of team-related knowledge: Design and validation of a shared mental model instrument, Human Resource Development International 10 (2007) 437–454.

[40] B. Kaiser, A. Csiszar, A. Verl, Generative models for direct generation of cnc toolpaths, in: 2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), IEEE, 2018, pp. 1–6.

[41] C. Metz, Meet gpt-3. it has learned to code (and blog and argue). (published 2020), 2022. URL: https://www.nytimes.com/2020/11/24/science/artificial-intelligence-ai-gpt3.html.

[42] S. I. Ross, F. Martinez, S. Houde, M. Muller, J. D. Weisz, The programmer's assistant: Conversational interaction with a large language model for software development, in: 28th International Conference on Intelligent User Interfaces, 2023.

[43] W. Geyer, L. B. Chilton, J. D. Weisz, M. L. Maher, Hai-gen 2021: 2nd workshop on human-ai co-creation with generative models, in: 26th International Conference on Intelligent User Interfaces-Companion, 2021, pp. 15–17.

[44] M. Muller, L. B. Chilton, A. Kantosalo, C. P. Martin, G. Walsh, Genaichi: Generative ai and hci, in: CHI Conference on Human Factors in Computing Systems Extended Abstracts, 2022, pp. 1–7.

[45] M. Muller, P. Agelov, H. Daume, Q. V. Liao, N. Oliver, D. Piorkowski, et al., Hcai@neurips 2022, human centered ai, in: Annual Conference on Neural Information Processing Systems, 2022.

[46] J. D. Weisz, M. L. Maher, H. Strobelt, L. B. Chilton, D. Bau, W. Geyer, Hai-gen 2022: 3rd workshop on human-ai co-creation with generative models, in: 27th International Conference on Intelligent User Interfaces, 2022, pp. 4–6.

[47] C. Aragon, S. Guha, M. Kogan, M. Muller, G. Neff, Human-Centered Data Science: An Introduction, MIT Press, 2022.

[48] D. Boyd, K. Crawford, Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon, Information, communication & society 15 (2012) 662–679.

[49] S. Costanza-Chock, Design justice: Community-led practices to build the worlds we need, The MIT Press, 2020.

[50] C. D'ignazio, L. F. Klein, Data feminism, MIT press, 2020.

[51] L. Gitelman, Raw Data is an Oxymoron, MIT Press, 2013.

[52] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, Advances in neural information processing systems 27 (2014).

[53] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[54] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (2020) 139–144.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[56] V. Chenthamarakshan, P. Das, S. C. Hoffman, H. Strobelt, I. Padhi, K. W. Lim, B. Hoover, M. Manica, J. Born, T. Laino, et al., Cogmol: target-specific and selective drug design for covid-19 using deep generative models, arXiv preprint arXiv:2004.01215 (2020).

[57] V. Chenthamarakshan, P. Das, I. Padhi, H. Strobelt, K. W. Lim, B. Hoover, S. C. Hoffman, A. Mojsilovic, Target-specific and selective drug design for covid-19 using deep generative models, 2020. arXiv:2004.01215.

[58] A. Liapis, G. N. Yannakakis, J. Togelius, et al., Sentient sketchbook: Computer-aided game level authoring., in: FDG, 2013, pp. 213–220.

[59] J. Rose, Facebook pulls its new 'ai for science' because it's broken and terrible, Vice (2022). URL: https://www.vice.com/en/article/3adyw9/facebook-pulls-its-new-ai-for-science-because-its-broken-and-terrible.

[60] B. Roziere, M.-A. Lachaux, L. Chanussot, G. Lample, Unsupervised translation of programming languages., in: NeurIPS, 2020.

[61] S. Kulal, P. Pasupat, K. Chandra, M. Lee, O. Padon, A. Aiken, P. S. Liang, Spoc: Search-based pseudocode to code, Advances in Neural Information Processing Systems 32 (2019).

[62] Github, Copilot, 2021. URL: https://copilot.github.com.

[63] B. Shneiderman, Human-centered artificial intelligence: Reliable, safe & trustworthy, International Journal of Human–Computer Interaction 36 (2020) 495–504.

[64] B. Shneiderman, Human-centered ai, Issues in Science and Technology 37 (2021) 56–61.

[65] V. Liu, L. B. Chilton, Neurosymbolic generation of 3d animal shapes through semantic controls., in: IUI Workshops, 2021.

[66] P. von Platen, How to generate text: using different decoding methods for language generation with transformers, Hugging Face Blog (2020). URL: https://huggingface.co/blog/how-to-generate.

[67] R. Cheng, R. Wang, T. Zimmermann, D. Ford, "it would work for me too": How online communities shape software developers' trust in ai-powered code generation tools, arXiv preprint arXiv:2212.03491 (2022).

[68] M. Scheutz, S. A. DeLoach, J. A. Adams, A framework for developing and using shared mental models in human-agent teams, Journal of Cognitive Engineering and Decision Making 11 (2017) 203–224.

[69] S. M. Fiore, E. Salas, J. A. Cannon-Bowers, Group dynamics and shared mental model development, How people evaluate others in organizations 234 (2001).

[70] J. E. Mathieu, T. S. Heffner, G. F. Goodwin, E. Salas, J. A. Cannon-Bowers, The influence of shared mental models on team process and performance., Journal of applied psychology 85 (2000) 273.

[71] M. Guzdial, N. Liao, J. Chen, S.-Y. Chen, S. Shah, V. Shah, J. Reno, G. Smith, M. O. Riedl, Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators, in: Proceedings of the 2019 CHI conference on human factors in computing systems, 2019, pp. 1–13.

[72] P. M. Fitts, M. Viteles, N. Barr, D. Brimhall, G. Finch, E. Gardner, W. Grether, W. Kellum, S. Stevens, Human engineering for an effective air-navigation and traffic-control system, and appendixes 1 thru 3, Technical Report, Ohio State Univ Research Foundation Columbus, 1951.

[73] T. B. Sheridan, W. L. Verplank, Human and computer control of undersea teleoperators, Technical Report, Massachusetts Inst of Tech Cambridge Man-Machine Systems Lab, 1978.

[74] R. Parasuraman, T. B. Sheridan, C. D. Wickens, A model for types and levels of human interaction with automation, IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans 30 (2000) 286–297.

[75] E. Horvitz, Principles of mixed-initiative user interfaces, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '99, Association for Computing Machinery, New York, NY, USA, 1999, p. 159–166. URL: https://doi.org/10.1145/302979.303030. doi:10.1145/302979.303030.

[76] V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilovic, et al., Ai explainability 360: An extensible toolkit for understanding data and machine learning models., J. Mach. Learn. Res. 21 (2020) 1–6.

[77] U. Ehsan, P. Wintersberger, Q. V. Liao, E. A. Watkins, C. Manger, H. Daumé III, A. Riener, M. O. Riedl, Human-centered explainable ai (hcxai): beyond opening the black-box of ai, in: CHI Conference on Human Factors in Computing Systems Extended Abstracts, 2022, pp. 1–7.

[78] Q. V. Liao, M. Singh, Y. Zhang, R. Bellamy, Introduction to explainable ai, in: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–3.

[79] A. Simkute, A. Surana, E. Luger, M. Evans, R. Jones, Xai for learning: Narrowing down the digital divide between "new" and "old" experts, in: Adjunct Proceedings of the 2022 Nordic Human-Computer Interaction Conference, 2022, pp. 1–6.

[80] C. Molnar, Interpretable machine learning, Lulu. com, 2020.

[81] M. Muller, A. Stroymayer, Forgetting practices in the data sciences, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022. In press.

[82] C. Pinhanez, Expose uncertainty, instill distrust, avoid explanations: Towards ethical guidelines for ai, in hcai@neurips 2021 workshop, 2021. URL: https://www.google.com/url?q=https%3A%2F%2Farxiv.org%2Fabs%2F2112.01281&sa=D.

[83] C. Pinhanez, Breakdowns, language use, and weird errors: Past, present, and future of research on conversational agents at brl, in ibm research cambridge lab guess speaker series, 2022.

[84] S. Lin, J. Hilton, O. Evans, Truthfulqa: Measuring how models mimic human falsehoods, arXiv preprint arXiv:2109.07958 (2021).

[85] S. Kantayya, Coded bias, 2020. URL: https://www.pbs.org/independentlens/documentaries/coded-bias/.

[86] K. Spiel, " why are they all obsessed with gender?"—(non) binary navigations through technological infrastructures, in: Designing Interactive Systems Conference 2021, 2021, pp. 478–494.

[87] A. Lyn, Risky business: Artificial intelligence and risk assessments in sentencing and bail procedures in the united states, Available at SSRN 3831441 (2020).

[88] D. Saxena, K. Badillo-Urquiola, P. J. Wisniewski, S. Guha, A framework of high-stakes algorithmic decision-making for the public sector developed through a case study of child-welfare, Proceedings of the ACM on Human-Computer Interaction 5 (2021) 1–41.

[89] S. Lindtner, S. Bardzell, J. Bardzell, Reconstituting the utopian vision of making: Hci after technosolutionism, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2016, pp. 1390–1402.

[90] M. A. Madaio, L. Stark, J. Wortman Vaughan, H. Wallach, Co-designing checklists to understand organizational challenges and opportunities around fairness in ai, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–14.

[91] A. Resseguier, R. Rodrigues, Ethics as attention to context: recommendations for the ethics of artificial intelligence, Open Research Europe 1 (2021) 27.

[92] B. Friedman, D. G. Hendry, Value sensitive design: Shaping technology with moral imagination, Mit Press, 2019.

[93] D. G. Hendry, B. Friedman, S. Ballard, Value sensitive design as a formative framework, Ethics and Information Technology 23 (2021) 39–44.

[94] J. K. Miller, B. Friedman, G. Jancke, B. Gill, Value tensions in design: the value sensitive design, development, and appropriation of a corporation's groupware system, in: Proceedings of the 2007 international ACM conference on Supporting group work, 2007, pp. 281–290.

[95] N. K. Denzin, Y. S. Lincoln, L. T. Smith, et al., Handbook of critical and indigenous methodologies, Sage, 2008.

[96] C. DiSalvo, Design as democratic inquiry: putting experimental civics into practice, MIT Press, 2022.

[97] G. R. Hayes, Knowing by doing: action research as an approach to hci, in: Ways of Knowing in HCI, Springer, 2014, pp. 49–68.

[98] J. Cho, A. Zala, M. Bansal, Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers, arXiv preprint arXiv:2202.04053 (2022).

[99] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al., Extracting training data from large language models, in: 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2633–2650.

[100] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, C. Zhang, Quantifying memorization across neural language models, arXiv preprint arXiv:2202.07646 (2022).

[101] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, et al., Large language models encode clinical knowledge, arXiv preprint arXiv:2212.13138 (2022).

[102] J. Albrecht, E. Kitanidis, A. J. Fetterman, Despite" super-human" performance, current llms are unsuited for decisions about ethics and safety, arXiv preprint arXiv:2212.06295 (2022).

[103] W. D. Heaven, Why meta's latest large language model survived only three days online, 2022. URL: https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/.

[104] J. Vincent, Ai-generated answers temporarily banned on coding q&a site stack overflow, 2022. URL: https://www.theverge.com/2022/12/5/23493932/chatgpt-ai-generated-answers-temporarily-banned-stack-overflow-llms-dangers.

[105] E. Meskys, J. Kalpokiene, P. Jurcys, A. Liaudanskas, Regulating deep fakes: legal and ethical considerations, Journal of Intellectual Property Law & Practice 15 (2020) 24–31.

[106] C. Stupp, Fraudsters used ai to mimic ceo's voice in unusual cybercrime case, The Wall Street Journal (2019). URL: https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402.

[107] S. Yang, K. Shu, S. Wang, R. Gu, F. Wu, H. Liu, Unsupervised fake news detection on social media: A generative approach, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 5644–5651.

[108] R. R. Morris, We provided mental health support to about 4,000 people — using gpt-3. here's what happened, 2023. URL: https://twitter.com/RobertRMorris/status/1611450197707464706.

[109] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots: Can language models be too big?, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 610–623.

[110] G. Franceschelli, M. Musolesi, Copyright in generative deep learning, Data & Policy 4 (2022).

[111] K. Hristov, Artificial intelligence and the copyright dilemma, Idea 57 (2016) 431.

[112] M. D. Murray, Generative and ai authored artworks and copyright law, Available at SSRN (2022).

[113] M. Butterick, Github copilot litigation, 2022. URL: https://githubcopilotlitigation.com.

[114] D. Wang, J. D. Weisz, M. Muller, P. Ram, W. Geyer, C. Dugan, Y. Tausczik, H. Samulowitz, A. Gray, Human-ai collaboration in data science: Exploring data scientists' perceptions of automated ai, Proceedings of the ACM on Human-Computer Interaction 3 (2019) 1–24.

[115] Q. Wang, K. Saha, E. Gregori, D. Joyner, A. Goel, Towards mutual theory of mind in human-ai interaction: How language reflects what students perceive about a virtual teaching assistant, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–14.

[116] A. Wilkins, Will ai text-to-image generators put illustrators out of a job?, NewScientist (2022). URL: https://www.newscientist.com/article/2322056-will-ai-text-to-image-generators-put-illustrators-out-of-a-job/.

[117] M. R. Morris, C. J. Cai, J. Holbrook, C. Kulkarni, M. Terry, The design space of generative models, in: Proceedings of the NeurIPS 2022 Workshop on Human-Centered AI (NeurIPS 2022), 2022.

[118] S. Kollmansberger, Helping students build a mental model of computation, in: Proceedings of the fifteenth annual conference on Innovation and technology in computer science education, 2010, pp. 128–131.