Beyond One-Size-Fits-All: How User Objectives Shape Counterfactual Explanations

Orfeas Menis Mastromichalakis, Jason Liartis and Giorgos Stamou

Artificial Intelligence and Learning Systems Laboratory, National Technical University of Athens

Abstract

Counterfactual Explanations (CFEs) have emerged as a powerful tool for interpreting machine learning models by illustrating alternative scenarios where key factors differ. Despite their widespread adoption, the existing literature often overlooks the diverse needs and objectives of users across various domains—ranging from guiding decision-making to understanding model behavior and assessing robustness—and their implications for CFE characteristics. As a result, CFEs frequently fail to adequately address these distinct use cases, leading to suboptimal and sometimes misleading explanations. In this paper, we advocate for a more user-centered approach to CFEs, emphasizing the importance of aligning their characteristics with user objectives. We identify three primary use cases—actionable user guidance, system understanding, and vulnerability assessment—and examine the desired properties of CFEs in each case. By addressing these differences, we aim to inform the design and deployment of more effective, context-aware explanations that meet the unique needs of users while ensuring the accuracy and usefulness of the insights provided.

1. Introduction

Explainable Artificial Intelligence (XAI) has become increasingly indispensable as AI systems are integrated into various facets of society. It aims to elucidate the opaque decision-making processes of machine learning algorithms, offering transparency and useful insights, and fostering trust among multiple stakeholders from AI engineers to end-users. Counterfactual Explanations, grounded in research from philosophy [1] and psychology on counterfactual thinking [2], explore why a particular outcome occurred by examining alternative scenarios where key factors or events were different. In artificial intelligence, these explanations provide insight into how alterations in input features would have influenced an AI system's output, thereby facilitating a deeper understanding of its behavior and decision-making mechanisms. Due to their contrastive nature, CFEs are close to how humans perceive explanations [3], facilitating an intuitive way to study a model's behavior.

While numerous existing works delve into the desired characteristics of counterfactual explanations [4, 5, 6, 7] and how to evaluate them [8], they often approach them with a unified strategy encompassing multiple objectives including detecting biases, providing actionable recourse, increasing trust, and enhancing understandability. However, this approach overlooks the fact that the diverse needs and objectives of users across various applications and

Late-breaking work, Demos and Doctoral Consortium, colocated with the 3rd World Conference on eXplainable Artificial Intelligence: July 09–11, 2025, Istanbul, Turkey

menorf@ails.ece.ntua.gr (O. Menis Mastromichalakis); gstam@cs.ntua.gr (G. Stamou)

domains necessitate different properties for counterfactual explanations [9]. Consequently, the explanations generated may fail to adequately address all use cases, as the requirements for one user's target could directly conflict with another's. This phenomenon of conflicting motivations and ambiguity of objectives in XAI has been thoroughly discussed in literature [10, 11]. Our work aligns with numerous works that call for contextualized design, development, and evaluation of explanations in terms of application [12], target audience [13, 14, 15, 16], and end-goal [17, 6]. Therefore, there is a pressing need to distinguish between the distinct use cases in which counterfactual explanations are applied. By doing so, we can pinpoint the specific desired characteristics for each use case, rather than attempting to create counterfactuals that aim to cover all scenarios, which is inherently infeasible.

In this paper, we advocate for a more nuanced understanding of counterfactual explanations, recognizing that their desired properties vary significantly depending on user objectives and target applications. Building on existing literature [5, 6], we identify three primary scenarios based on user objectives and examine the key properties that CFEs should exhibit in each case. The first case concerns *actionable user guidance*, where CFEs assist users in decision-making by providing actionable guidance on how to modify inputs to achieve a specific desired outcome. The second is *system understanding*, a central focus in explainable AI, where CFEs are used to analyze model behavior across real-world inputs, offering insights into the model's biases and intricate decision-making mechanisms. The third is *vulnerability assessment*, which leverages CFEs to identify weaknesses in a model, such as susceptibility to input perturbations, assessing its robustness. While CFEs can support all these objectives, their effectiveness depends on how well they align with the specific needs of each use case. For instance, CFEs designed to guide user actions may fail to reveal deeper insights into a model's reasoning, whereas those tailored for system investigation may lack actionable recommendations. By acknowledging these distinctions, we can develop more targeted and effective explanations that empower users across different contexts, ultimately fostering more meaningful human-AI collaboration.

2. Preliminaries

In this section, we explain and briefly discuss some concepts that we use throughout this paper. We assume that there is an AI system f under examination that accepts an input x and produces an output $f(x) = y$. For example, the AI system might be an automated loan approval system deployed by a bank. In this case, x would be a client's application that might include information such as age, gender, occupation, income, number of existing loans, etc. and y would be the AI system's decision, approve or reject. A *counterfactual instance* of x is a new input $x'(\neq x)$ that produces a different output $y'(\neq y)$. x' is also generally expected to minimize some metric μ among all inputs that produce a different output, i.e., $x' = \underset{z}{\operatorname{argmin}}\{\mu(x, z) \mid f(z) \neq y\}$. We further discuss this in the following paragraph. The system that produces counterfactuals instances, may be referred to as the *counterfactual explainer* or *counterfactual editor*. The differences between x and x' are also referred to as the *counterfactual edit*. The counterfactual edit indicates what changes need to be made to the input to receive a different output. In the automated loan approval example, a counterfactual edit may suggest to a user with a rejected application, that they need to change occupation or pay off one of their existing loans for their application to be approved.

Although, in principle, any instance $x' \neq x$ with $y' \neq y$ can be considered as a counterfactual, most approaches require x' to be close to x by some metric μ, which offers two key benefits. First, it ensures that the counterfactual edit—the differences between x and x'—is minimal, making it easier to interpret which specific changes influence the model's output. This helps isolate the impact of different input features. Second, it transforms the search for a suitable counterfactual into an optimization problem, where the objective is to find the closest possible x' with $y' \neq y$ leveraging heuristics or established optimization techniques. It is crucial to note that the choice of metric used to calculate the distance between x and x' is not merely a technical detail; it fundamentally shapes the characteristics of CFEs and is closely tied to the different scenarios and use cases analyzed in this paper.

Although valid, some counterfactual edits may lead to instances that are highly uncommon or even unattainable in the real world. For example, it is rare for a teacher to earn an income of $1 million, and it is impossible for someone to have an age of -1. Therefore, assessing whether a counterfactual instance is coherent with a reference population—such as real-world distributions—is crucial, as its plausibility can significantly impact its usefulness in different contexts. Counterfactual instances that align with a reference population are referred to as *plausible* or *feasible* [18, 6]. Plausible instances are also said to be close to the data manifold [7], or in-distribution, while implausible instances are said to be far from the data manifold and out-of-distribution (OOD).

However, even if a counterfactual instance is plausible, it may still be unattainable from the original instance due to constraints on the actions required to achieve it. For example, while any date could be a valid birth date, an individual cannot change their own birth date. Thus, it is important to assess whether a plausible counterfactual instance can actually be reached from the original input through real-world actions. Counterfactual edits that modify only mutable features are referred to as *actionable* [6]. An actionable edit consists of a set of feasible changes that can be applied to the input instance to attain the counterfactual instance. While some features, such as date of birth, are inherently immutable, others, like occupation or income, may be modifiable in principle, though the feasibility of such modifications can vary across individuals. Given this variability, it is desirable for an explainer to account for user-specific constraints when determining actionability, ensuring that counterfactual suggestions align with real-world possibilities.

The literature discusses various properties and characteristics beyond plausibility and actionability that also influence the effectiveness and applicability of CFEs. For instance, counterfactual edits that involve modifying only a few features are referred to as *sparse* [7], a property often desired for clarity and interpretability. However, in this work, we focus primarily on actionability and plausibility, as these are the key differentiating factors among the three scenarios analyzed in the following section.

Table 1
Desirable properties for each use case.

Use Case	Plausibility	Actionability
Actionable User Guidance	✓	✓
System Understanding	✓	
Vulnerability Assessment		

3. Use Cases and User Objectives

In this section, we analyze three distinct user-objective-driven use cases of counterfactual explanations, focusing on the properties of actionability and plausibility. Each use case represents a real-world application scenario with different goals and constraints. Specifically, we examine three key scenarios:

1. when an end-user seeks guidance on achieving a desired outcome,
2. when a user aims to investigate and understand an AI system's behavior—such as detecting biases, flaws, or inconsistencies on real distributions,
3. and when a user attempts to identify system vulnerabilities and strengthen defenses against potential attacks.

As demonstrated in Table 1, actionability and plausibility are not universally desirable properties. Instead, their relevance depends on user objectives, as they may sometimes impose constraints that limit insights or hinder the effectiveness of the explanation. It is important to note that the absence of a checkmark in the table does not imply that counterfactuals with these properties are excluded. Rather, it reflects an opt-in approach, where a checkmark indicates that all counterfactuals in that use case must exhibit the given property, rather than an opt-out approach, where their absence would mean such counterfactuals should never be considered. For instance, while plausible and actionable counterfactuals remain relevant in vulnerability assessment, relying only on them would overlook important failure modes of the system. As analyzed in the following subsections, the issue in later use cases arises when counterfactuals are constrained to those exhibiting these properties, not from their presence itself.

3.1. Actionable User Guidance

In this use case, also known as *(actionable) recourse* [5, 19, 20], the end-user seeks guidance on modifying input features to achieve a desired outcome. For instance, in the automated loan approval system example, a bank customer whose application was rejected may want to know what changes could increase their chances of approval. This is the most restrictive use case, as it requires counterfactual explanations to be both actionable and plausible. Actionability ensures that suggested modifications involve only mutable features, while plausibility guarantees that these modifications are realistic within the reference population. Providing impractical suggestions—such as changing one's birthplace or attaining an impossible state, like a negative age—would make the CFE unhelpful, as the user would be unable to act on it. The goal is to offer meaningful and feasible recommendations that align with real-world constraints.

The choice of minimality metric also plays a crucial role in optimizing CFEs for this use case. Some works advocate for the use of a sparsity-inducing norm such as l_0 or l_1, since it provides the user with a small set of specific goals [21, 22]. Sometimes, these norms result in a counterfactual edit with a big change to a single feature, which might be less attainable than a few smaller changes. Other norms such as l_2 and l_∞ along with a sparsity-inducing penalty might be more suitable. A different approach focuses on the cost of real-world actions. Instead of evaluating minimality with regard to differences between the two different instances x and x', they work on a higher-level space of actions that a user needs to take in order to arrive at x' starting from x, simultaneously addressing issues of cause and effect [23, 24, 25]. In the example of the loan approval system, a counterfactual edit suggesting a change of occupation may be irrelevant if it does not refer to a set of actions necessary to change occupations and ignores that other features may have to change as a result of those actions, such as income and education level.

3.2. System Understanding

In this use case, the user seeks to analyze and understand the behavior of the AI system within the domain of in-distribution data. Counterfactual explanations help uncover potential biases and inconsistencies in the model's decision-making. In this context, plausibility is crucial because the goal is to analyze the model's decision-making within realistic, real-world distributions. Allowing non-plausible counterfactuals could introduce noise and non-sensical edits, potentially obscuring meaningful biases among irrelevant or unrealistic changes. Conversely, if we were to restrict ourselves only to actionable edits, we would miss biases related to immutable features—such as gender or ethnicity—which are often at the core of discriminatory behavior in AI systems. Many real-world biases manifest in decisions that treat individuals differently based on characteristics they cannot change, so excluding counterfactuals that vary these attributes would prevent us from fully diagnosing such issues. For instance, an AI engineer developing a loan approval or candidate screening system may investigate whether the model exhibits biases related to sensitive attributes such as gender or skin color. The engineer may also assess whether the model's decisions align with human intuition and logical reasoning. For example, if a counterfactual edit suggests that reducing one's income would increase the likelihood of loan approval, it would be counter-intuitive, possibly indicating an unexpected flaw in the system's behavior. Presenting a multitude of counterfactuals is very important in this use case since their contrast can reveal biases. For instance, one counterfactual edit might suggest an increase of income by $10,000 to secure loan approval, while another might suggest an increase of income by $1,000 and a change of gender. This looser income requirement for a different gender would reveal a potential gender bias.

In such cases, the choice of minimality metric depends on the focus of the investigation. When stress-testing for specific biases, it is often beneficial to isolate certain sensitive features or heavily penalize non-sensitive alterations. This can be achieved through proper customization of the minimality metric. Conversely, when the goal is a broader understanding of model behavior, norm-based metrics such as l_1 or l_2 are commonly used.

3.3. Vulnerability Assessment

In this use case, the user aims to identify potential weaknesses or vulnerabilities in the AI system. Counterfactual explanations serve as a tool to assess the model's robustness against small perturbations or out-of-distribution inputs. For instance, a security engineer may want to test whether slight modifications to input data—such as leaving fields empty, providing invalid values, or introducing minor inconsistencies—could compromise the system's integrity. The primary emphasis here lies on robustness, where considerations of plausibility and actionability pose potential conflicts with the user's objectives as they could hinder the detection of vulnerabilities involving random noise or out-of-distribution permutations. Imperceptible changes to the input that significantly alter the output, commonly known as adversarial examples [26] also fall under this category. Although typically treated as a distinct concept, sometimes explicitly suppressed by counterfactual explainers, they are, in essence, a specific case of counterfactual edits. Rather than viewing them as a separate phenomenon, it is preferable to specify whether a given counterfactual explainer accommodates this use case.

For such robustness assessments, norm-based metrics are generally preferred for minimality, as they focus on small perturbations that influence the output without necessarily having semantic meaning. Counterfactual edits with low norm minimality are particularly useful, as they identify points on the data manifold that are very close to the original input, allowing for a fine-grained evaluation of the system's resilience. However, the choice of norm metric still impacts the nature of the generated counterfactuals. For example, in the domain of image classifiers, some editors employ the l_0 norm to change a single, or very few, pixels [27], while others use the l_2 norm to produce a set of many tiny changes that are imperceptible to the human eye [28]. Both are valid approaches with different implications for robustness testing. To comprehensively assess a system's vulnerabilities, a variation of such metrics should be considered to ensure that different types of adversarial weaknesses are adequately explored.

4. Conclusions

In this paper, we underscore the importance of a nuanced understanding of counterfactual explanations. By recognizing the variability in desired properties based on user objectives and target applications, we have advocated for a tailored approach to the design and development of CFEs. Our analysis of three main user objectives and their relation to the key concepts of plausibility and actionability has revealed that the desired characteristics of CFEs differ significantly depending on the end task, highlighting the necessity of considering user needs in the explanation process. Through this study, we have demonstrated the limitations of a one-size-fits-all approach to CFEs and emphasized the need for customized explanations that address the specific requirements of users across diverse scenarios.

Acknowledgments

Funding for the work was received from the European Commission Directorate-General for Communications Networks, Content and Technology grant no. 101135809 (EXTRA-BRAIN).

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

[1] D. Lewis, Counterfactuals, John Wiley & Sons, 2013.

[2] R. M. Byrne, Counterfactual thought, Annual review of psychology 67 (2016) 135–157.

[3] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1–38. URL: https://www.sciencedirect.com/science/article/pii/S0004370218305988. doi:10.1016/j.artint.2018.07.007.

[4] S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, C. Shah, Counterfactual explanations and algorithmic recourses for machine learning: A review, arXiv preprint arXiv:2010.10596 (2020).

[5] A. Karimi, G. Barthe, B. Schölkopf, I. Valera, A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. corr abs/2010.04050 (2020), publications at ACM Computing Surveys (2020).

[6] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Mining and Knowledge Discovery (2022) 1–55.

[7] S. Verma, J. Dickerson, K. Hines, Counterfactual explanations for machine learning: Challenges revisited, arXiv preprint arXiv:2106.07756 (2021).

[8] G. Filandrianos, E. Dervakos, O. Menis-Mastromichalakis, C. Zerva, G. Stamou, Counterfactuals of counterfactuals: a back-translation-inspired approach to analyse counterfactual editors, arXiv preprint arXiv:2305.17055 (2023).

[9] O. M. Mastromichalakis, J. Liartis, G. Stamou, Beyond one-size-fits-all: Adapting counterfactual explanations to user objectives, arXiv preprint arXiv:2404.08721 (2024).

[10] Z. C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery., Queue 16 (2018) 31–57. URL: https://doi.org/10.1145/3236386.3241340. doi:10.1145/3236386.3241340.

[11] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215. doi:10.1038/s42256-019-0048-x.

[12] M. Norkute, Ai explainability: Why one explanation cannot fit all, in: ACM CHI Workshop on Operationalizing Human-Centered Perspectives in Ex-plainable AI (HCXAI), 2021.

[13] K. Sokol, P. Flach, One explanation does not fit all: The promise of interactive explanations for machine learning transparency, KI-Künstliche Intelligenz 34 (2020) 235–250.

[14] J. Liartis, E. Dervakos, O. Menis-Mastromichalakis, A. Chortaras, G. Stamou, Searching

for explanations of black-box classifiers in the space of semantic queries, Semantic Web (2023) 1–42.

[15] O. M. Mastromichalakis, E. Dervakos, A. Chortaras, G. Stamou, Rule-based explanations of machine learning classifiers using knowledge graphs, in: Proceedings of the AAAI Symposium Series, volume 3, 2024, pp. 193–202.

[16] S. Dhanorkar, C. T. Wolf, K. Qian, A. Xu, L. Popa, Y. Li, Who needs to know what, when?: Broadening the explainable ai (xai) design space by looking at explanations across the ai lifecycle, in: Proceedings of the 2021 ACM Designing Interactive Systems Conference, 2021, pp. 1591–1602.

[17] C. T. Wolf, Explainability scenarios: towards scenario-based xai design, in: Proceedings of the 24th International Conference on Intelligent User Interfaces, 2019, pp. 252–257.

[18] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, P. Flach, Face: feasible and actionable counterfactual explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 344–350.

[19] P. M. VanNostrand, D. M. Hofmann, L. Ma, E. A. Rundensteiner, Actionable recourse for automated decisions: Examining the effects of counterfactual explanation type and presentation on lay user understanding, in: Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’24, Association for Computing Machinery, New York, NY, USA, 2024, p. 1682–1700. URL: https://doi.org/10.1145/3630106.3658997. doi:10.1145/3630106.3658997.

[20] J. Yetukuri, I. Hardy, Y. Liu, Towards user guided actionable recourse, in: Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society, AIES ’23, Association for Computing Machinery, New York, NY, USA, 2023, p. 742–751. URL: https://doi.org/10.1145/3600211.3604708. doi:10.1145/3600211.3604708.

[21] C. Russell, Efficient search for diverse coherent explanations, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* ’19, Association for Computing Machinery, New York, NY, USA, 2019, p. 20–28. URL: https://doi.org/10.1145/3287560.3287569. doi:10.1145/3287560.3287569.

[22] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Harv. JL & Tech. 31 (2017) 841.

[23] R. Crupi, A. Castelnovo, D. Regoli, B. San Miguel Gonzalez, Counterfactual explanations as interventions in latent space, Data Mining and Knowledge Discovery 38 (2024) 2733–2769.

[24] G. Ramakrishnan, Y. C. Lee, A. Albarghouthi, Synthesizing action sequences for modifying model decisions, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 5462–5469. doi:10.1609/aaai.v34i04.5996.

[25] A.-H. Karimi, B. Schölkopf, I. Valera, Algorithmic recourse: from counterfactual explanations to interventions, 2020. URL: https://arxiv.org/abs/2002.06278. arXiv:2002.06278.

[26] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).

[27] J. Su, D. V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Transactions on Evolutionary Computation 23 (2019) 828–841.

[28] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.