Praxa: A Framework for What-If Analysis

SNEHA GATHANI, University of Maryland, College Park, USA
KEVIN LI, University of Maryland, College Park, USA
RAGHAV THIND, University of Maryland, College Park, USA
SIRUI ZENG, University of Maryland, College Park, USA
MATTHEW XU, University of Maryland, College Park, USA
PETER J. HAAS, University of Massachusetts, Amherst, USA
ÇAĞATAY DEMIRALP, AWS AI Labs, USA and MIT CSAIL, USA
ZHICHENG LIU, University of Maryland, College Park, USA

Various analytical techniques–such as scenario modeling, sensitivity analysis, perturbation-based analysis, counterfactual analysis, and parameter space analysis–are used across domains to explore hypothetical scenarios, examine input–output relationships, and identify pathways to desired results. Although termed differently, these methods share common concepts and methods, suggesting unification under *what-if analysis*. Yet a unified framework to define motivations, core components, and its distinct types is lacking. To address this gap, we reviewed 141 publications from leading visual analytics and HCI venues (2014–2024). Our analysis (1) outlines the motivations for what-if analysis, (2) introduces Praxa, a structured framework that identifies its fundamental components and characterizes its distinct types, and (3) highlights challenges associated with the application and implementation. Together, our findings establish a standardized vocabulary and structural understanding, enabling more consistent use across domains and communicate with greater conceptual clarity. Finally, we identify open research problems and future directions to advance what-if analysis.

CCS Concepts: • **Human-centered computing** → **Visualization theory, concepts and paradigms**; **Empirical studies in visualization**; *Interactive systems and tools*; *HCI theory, concepts and models*; • **Computing methodologies** → **Machine learning**.

Additional Key Words and Phrases: What-if Analysis, Scenario Modeling, Sensitivity Analysis, Inverse Modeling, Counterfactual Analysis, Parameter Space Analysis, Literature Review Analysis

ACM Reference Format:
Sneha Gathani, Kevin Li, Raghav Thind, Sirui Zeng, Matthew Xu, Peter J. Haas, Çağatay Demiralp, and Zhicheng Liu. 2018. Praxa: A Framework for What-If Analysis. In *Proceedings of Conference on Human Factors in Computing Systems (CHI '26).* ACM, New York, NY, USA, 30 pages. https://doi.org/XXXXXXX.XXXXXXX

Authors' Contact Information: Sneha Gathani, sgathani@umd.edu, University of Maryland, College Park, College Park, Maryland, USA; Kevin Li, kli36@terpmail.umd.edu, University of Maryland, College Park, College Park, Maryland, USA; Raghav Thind, rthind@terpmail.umd.edu, University of Maryland, College Park, College Park, Maryland, USA; Sirui Zeng, szeng124@umd.edu, University of Maryland, College Park, College Park, Maryland, USA; Matthew Xu, mxu0601@terpmail.umd.edu, University of Maryland, College Park, College Park, Maryland, USA; Peter J. Haas, phaas@cs.umass.edu, University of Massachusetts, Amherst, Amherst, Massachusetts, USA; Çağatay Demiralp, cagatay@csail.mit.edu, AWS AI Labs, New York, New York, USA and MIT CSAIL, Cambridge, Massachusetts, USA; Zhicheng Liu, leozcliu@umd.edu, University of Maryland, College Park, College Park, Maryland, USA.

1 Introduction

Across diverse domains—including business, economics, healthcare, and urban planning—various analytical methods are employed to explore data relationships, interpret model behavior, identify alternative strategies for achieving desired outcomes, and support decision-making. These methods are described using different terminologies, such as scenario modeling [20, 53, 61], sensitivity analysis [51, 54], perturbation-based analysis [55], what-if analysis [30, 32, 52, 74, 76], inverse modeling [63, 65, 70], counterfactual analysis [45, 102], and parameter space analysis [12, 88]. For example, scenario modeling involves simulating selected potential conditions of factors to evaluate different outcomes [61, 99]. Sensitivity analysis assesses the impact of variations in specific factors on outcomes [9, 25]. Perturbation-based analysis deliberately introduces variations to factors and study system behavior [55]. What-if analysis identifies and optimizes influential factors to achieve desired outcomes [30, 32]. Inverse modeling determines the input factors required to produce specific outcomes [63], whereas counterfactual analysis identifies minimal changes in input factors necessary to change the model predictions [78]. Parameter space analysis systematically examines input variations to observe their effects on outputs [75].

Despite their diverse applications, these seemingly distinct analyses share common conceptual structures and objectives: they all involve exploring hypothetical scenarios to observe potential outcomes or identifying data and model configurations that achieve desired outcomes. For the rest of the paper, we adopt *what-if analysis* as an umbrella term for this family of methods, aligning with the terminology commonly adopted in commercial business intelligence tools [16, 26, 83, 87]. The methodological overlaps among these different what-if analyses suggest the potential of developing an underlying framework independent of specific domain applications and implementations. However, no work has systematically examined how these analyses relate to one another within a cohesive structure. Specifically, we identify three critical gaps in the current understanding and interpretation of these analyses. First, while different what-if analysis methods address different objectives across domains, the motivations underlying their use have not been articulated clearly in a domain-agnostic manner. Second, there has been insufficient analysis of the fundamental building blocks of what-if analysis—what elements to modify (e.g., data attributes or model parameters) and how to execute the modifications (e.g., dataset scoping, variable perturbation and constraint application, or reweighting in models). This lack of clarity makes it difficult to determine whether the different analyses share core components and to implement these analyses systematically. Third, distinguishing between different types of what-if analysis is often challenging. For example, inverse modeling and counterfactual analysis both trace outcomes back to their inputs, but differ in their implementation strategies, raising questions about whether they constitute distinct methodologies or are merely variations of a common analysis method. These ambiguities hinder systematic advancement in what-if analysis and complicate the development of generalizable analytical tools.

We address these gaps by performing an extensive literature review and analysis of 141 research publications from 2014-2024 across leading visual analytics and HCI venues including VIS, EuroVis, CHI, IUI, UIST, and TVCG. Specifically, we introduce Praxa, a framework for what-if analysis with the following contributions:

- We articulate the major motivations driving existing works to employ what-if analysis across different domains, ranging from understanding and debugging model behavior to incorporating user preferences and constraints (Section 3.1).
- We identify the building blocks (i.e., `components`) of what-if analysis along two fundamental dimensions: *what* key `dataset` and `model` elements constitute what-if analysis (Section 3.2), and *how* the analyses are conducted in terms of `user_operations` and `system_operations` (Section 3.3). Building on these `components`, we define and

characterize four distinct **TYPES** of what-if analysis, each with unique underlying assumptions and methodologies represented by combining different components (Section 3.4).

- We highlight common challenges in implementing and using what-if analysis discussed in the literature, as well as extend them to research gaps and opportunities to guide future research in what-if analysis (Section 7).

PRAXA establishes a standardized vocabulary for the components of different types of what-if analysis, providing a conceptual framework that allows researchers and practitioners to implement what-if analysis consistently in domains. PRAXA can serve as a practical guide for both researchers and practitioners. For instance, for researchers, PRAXA highlights underexplored combinations of components (e.g, plugging predictions of input variable values from **GOAL SEEK** back into **SENSITIVITY** to continue analysis), opening opportunities to develop new techniques and develop systems that extend current what-if analyses. For practitioners in different domains (e.g., marketing budgets [30, 31], healthcare interventions [5, 33, 46], urban policies [28], etc.), PRAXA provides explicit scaffolding of variables, constraints, and objectives onto a pre-defined set of `components`. This provides the breakdown of complex analyses into concise and actionable structure that is easier to communicate, audit, and reproduce, providing a foundation for building generalizable tools that can be specialized through modular adaptation to different domains.

2 Methodology

To articulate the motivations for employing what-if analysis, define its building blocks and types, synthesize the challenges in its implementation and use, and identify open research questions and future opportunities, we conducted a literature review of research publications on what-if analysis and related concepts. Figure 1 provides an overview of our nine-month process, including paper collection, analysis, and discussion of findings.

2.1 Paper Selection

We began by conducting a comprehensive keyword search for papers published between 2014 and 2024 in leading venues for visual analytics and HCI, including InfoVis, VAST, SciVis (later unified as VIS), EuroVis, CHI, IUI, UIST, and TVCG. The search keywords were designed to capture a broad range of terms related to what-if analysis, such as "scenario-based analysis", "sensitivity", "counterfactual", "perturbation", "what-if", "alternatives", "inverse modeling", "goal seek", and "parameter space". We also included partial terms like "scenario", "perturb", and "parameter" in our query to ensure

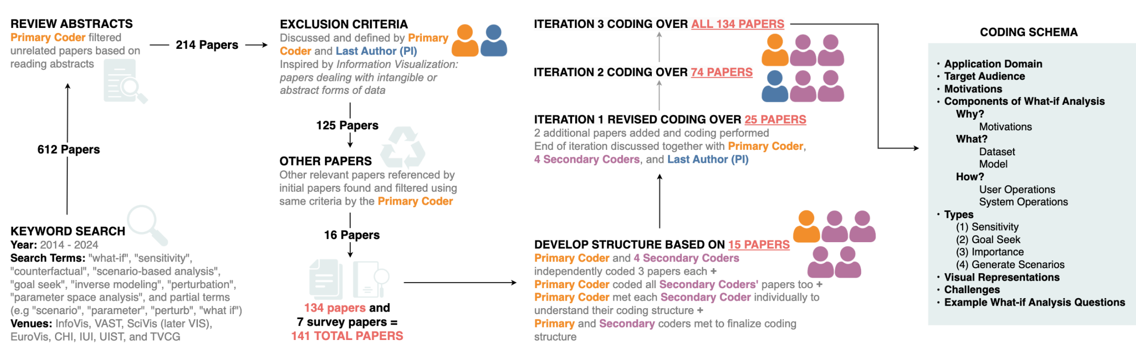


Fig. 1. Methodology of the paper selection, codebook development, and coding and analysis followed for the literature review.

comprehensive coverage of the relevant literature. We provide the complete list of keywords used in the supplementary materials [1]. This initially yielded 612 papers.

The first author (primary coder) reviewed the abstracts of all these papers to assess their relevance to what-if analysis. Papers that were deemed irrelevant, such as the papers dealing with scientific visualizations, theoretical discussions, or system optimization without user interaction were excluded. This process narrowed the selection down to 214 papers. Next, the primary coder and the last author (PI) collaboratively defined the inclusion and exclusion criteria. We focused specifically on papers dealing with intangible or abstract forms of data, thereby excluding works that centered on tangible or physical datasets (i.e., those involving physical equipment or hardware [64, 94] were excluded). This filtering process further narrowed the pool to 125 papers. Lastly, we employed a snowballing approach, identifying additional relevant papers from the references of this pool of 125 papers. Applying the same criteria, we included 16 more papers, bringing the total number of papers to review to 141. Of these papers, 7 were survey papers and excluded from the analysis. We conducted a deep analysis of the rest of the 134 papers to achieve our contributions.

2.2 Coding and Analysis Procedure

The coding and analysis involved a systematic and iterative process with six authors (first author as primary coder, four secondary coders, and the last author as the PI). This collaborative approach ensured the coding and analysis was robust, comprehensive, and consistent across all the papers.

[1] Supplementary materials URL: https://tinyurl.com/PraxaSupplementary

P92. What-If Tool: Interactive Probing of Machine Learning Models
Year: 2019
Searched Term: "what-if"
Venue: TVCG

Schema Structure	Analysis Codes
Application Domain	*Machine learning*
Target Audience	*Data scientists, data analysts, and domain experts*
Motivations	*To inspect, analyze, and understand ML systems*
What-if Analysis Method	*1. Data point editing* *2. Counterfactual reasoning*
Definition	*1. edit, add, or delete individual feature values or entire features within an instance and see effects on model prediction for that data point* *2. comparing differences to data points upon which the model predicted a different outcome*
Dataset Type	Tabular
Input Variables	*Capital Gain, Age, Hours-Per-Week, etc.*
Output Variable	*Income*
Model	*SVMs* and *NNs*
Model Tasks Supported	Regression and classification
Model Creation Process	Trained by authors
Relationship between Input and Output Variable	1. change individual data points values 2. get input variable values for flipped loan prediction
Data Features Created?	-
Data Subset Supported?	Yes, both *local subpopulations of data* and *global complete data*
Model Parameters Changed?	-
Any Values Changed?	-
Any User Needs Supported?	-
Visual to Conduct What-if Analysis	1. *Editing tables*
Visual to Observe Predictions from What-if Analysis	1. *Textual explanations, scatterplots,* and *bar graphs* showing changing predictions 1. *Partial dependency plots* to capture effects of change in features across a range 2. *Tables*
Example What-if Analysis Questions	*Would person X have gotten a loan if she were male instead of female?* *How much would a small increase in person X's income have affected the result?* ...
Challenges of What-if Analysis	*Constant changes were hard to keep track of and confusing*

Fig. 2. Example of selected schema and codes of the Wexler et al. [100] paper. The colors illustrate how various schema properties captured contribute to the different building blocks of PRAXA: `dataset`, `model`, `motivations`, `user_operations`, and `system_operations`.

The process began with five coders (primary and secondary) independently analyzing three distinct papers each, covering different what-if analysis methods (15 papers in total). Each coder identified important features to extract from the papers, such as the motivation behind the analysis, its definition, dataset type, input variables, output variables, models used, task performed, visuals adopted for providing input and observing the output, and example what-if analysis questions, which collectively formed the initial schema that guided the analysis of each paper. To strengthen the schema used across all coders, the primary coder re-coded the twelve papers analyzed by the secondary coders and met each secondary coder in one-on-one discussions to understand the features in their schema. After the individual discussions, all five coders met to consolidate their schemas, clarify definitions, and refine codes (i.e., predefined values) to ensure consistent granularity across all coders. For example, the 'target audience' feature was coded into four categories reflecting the primary role emphasized in the papers: domain experts (specialists directly related to the subject matter), data professionals (such as data scientists, analysts, and software engineers), researchers (those conducting studies or experiments within the field), and end-users (the individuals or organizations who ultimately use the product or service). The finalized schema, which outlined the key features to be captured from each of the papers, and the codebook, which listed and detailed the decided codes for each feature, formed the foundation for coding the remaining papers.

In the first iteration, we coded 25 papers (revised 15 previously coded ones + an additional 10) using the finalized schema. After completing this initial set, all coders and the PI, met to discuss the schema features and codebook, resolve confusion, and ensure consistency. This collaborative review process was repeated at the end of each iteration. The PI provided high-level perspective to keep codes not overly detailed nor excessively high-level. In the second iteration, we coded 49 additional papers (i.e., 74 total papers). Following this iteration, we further refined the schema and codebook based on discussions and feedback to enhance clarity and granularity. Finally, in the third iteration, the codebook was revised to cover all 134 papers. The iterative coding approach ensured consistency and reliability across all the reviewed papers. The final codebook with detailed descriptions is provided in the supplementary materials [1].

Each coder coded a distinct set of 27 papers (including those used to develop the schema and codebook), with the primary coder re-coding additional papers throughout the process as a quality check. Coding each paper took 1.5–2 hours, involving deep reading of the paper to map relevant information onto the schema. Figure 2, illustrates this process using Wexler et al.'s paper on "what-if analysis" [100] as an example. This paper introduces the What-If Tool, which enables data scientists, analysts, and domain experts to inspect, analyze, and understand machine learning models through interactive what-if analysis by requiring minimal coding.

After coding all the papers individually, the coders cross-analyzed a third of each other's papers in pairs to ensure reliability. Following individual and paired coding, the primary and secondary coders convened to synthesize the paper-level codes related to 'motivations', 'what-if analysis method', 'definitions', and 'challenges of what-if analysis', into broader high-level themes. These discussions took place over eight to ten virtual sessions (1.5 to 2.5 hours each), using digital affinity diagramming to cluster the paper-level codes into broad themes in Google Spreadsheets. Additionally, three to five hour-long discussions were held with all coders to discuss the review findings, clarify any ambiguities in coding, and refine the themes.

3 The PRAXA Framework

In this section, we introduce the PRAXA framework that addresses the ***why***, ***what***, and ***how*** perspectives of what-if analysis, as illustrated in Figure 3. We begin by describing ***why*** the reviewed works employed what-if analysis in terms of their *high-level motivations* with accompanying examples (Section 3.1). Next, we discuss ***what*** the *building blocks* of these analyses are (Section 3.2). We then move on to explain ***how*** the *operations* applied to the building blocks accomplish what-if analysis (Section 3.3). Finally, we formulate various types of what-if analysis, formalizing how each analysis type is accomplished by combining the building blocks and operations (Section 3.4).

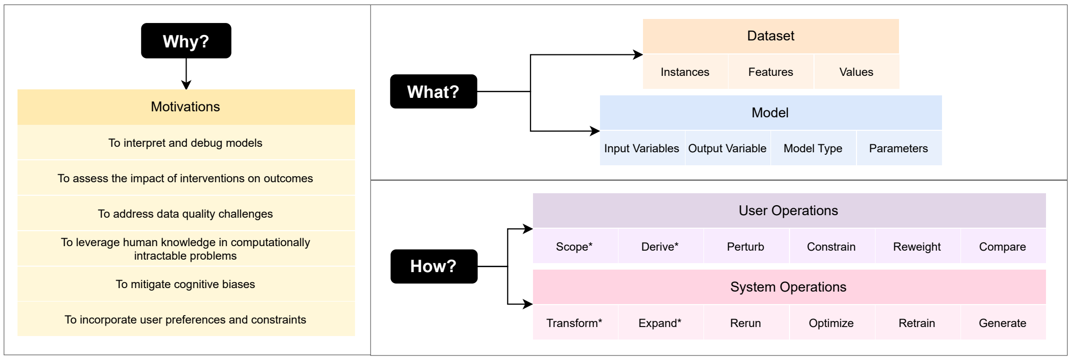


Fig. 3. The components of the PRAXA framework for what-if analysis are organized around three key dimensions identified through our literature review: ***why***, ***what***, and ***how***. The ***why*** dimension captures the underlying `motivations` for performing what-if analysis. The ***what*** dimension defines its core building blocks: the `dataset` and the `model`. The ***how*** dimension describes the operational mechanisms through which what-if analysis is performed, comprising both `user_operations` and `system_operations` that act upon the ***what*** components. The operations marked with * represent more general data analysis operations that, while not exclusive to what-if analysis, are frequently observed and play a key role in enabling it.

3.1 Why?

Here, we present the high-level `motivations` discussed in the papers for employing what-if analysis. If a paper discussed multiple motivations, we coded each explicitly, so the motivations are not mutually exclusive. Alongside examples of different what-if analysis methods for each motivation, we highlight overlaps (e.g., different methods described as the same) and inconsistencies (e.g., similar methods given different names). This underscores the need for a systematic understanding of *what* components define what-if analysis methods and *how* they can be combined to characterize distinct *types* of what-if analysis.

To interpret and debug models (29.8%). The reviewed literature often described predictive models as "black boxes" [5, 23, 96, 108]. This led to challenges in interpreting how models arrive at particular predictions [51, 96] and debugging them to identify and fix unexpected behaviors [39, 95].

Examples. Evirgen et al. [25] investigate text-to-image models by removing keywords from text prompts and measuring their impact on generated images. Such "sensitivity-based explanations" help provide insight into the model's behavior, robustness, and invariance properties by revealing which inputs shape the model's outputs. Hohman et al. [42] support model interpretability through "counterfactuals", which observe how modifying input features affects model predictions, and "feature importance", which ranks the features that most influence the overall prediction outcomes. Roberts and Tsiligkaridis [71] explore the interpretability of image classification models by "perturbing inputs" and observing changes in predictions. They also introduce "counterfactuals", showing how minor image modifications alter classifications to support model debugging. Yamanaka et al. [107] get Pareto front "optimal solutions" that give better model performance. Hence, many different what-if analyses are used to probe different aspects of the model and in turn, understand and debug their behavior.

To assess the impact of interventions on outcomes (17.16%). In contrast to the previous model-centric motivation, some reviewed works adopt an outcome-centric perspective, using what-if analysis to simulate and assess the effects of deliberate interventions or changes in input variables or model parameters on the model's predicted outcomes. By systematically observing how the model's output responds to various interventions, these works quantify the impact of various interventions, thereby supporting scenario evaluation and guiding decisions about which actions to implement.

Examples. Wu et al. [103] used "what-if analysis" to alter user opinions and predict sentiment shifts in tweets, while Luo et al [60] applied "sensitivity analysis" to understand how changes in climate features like precipitation and emissions impact water demand. Crisan and Correll [19] examined the "sensitivity of topic modeling" by removing stop terms or adding text, and Luboschik et al. [58] let analysts test "parameter configurations" to study their influence on protein movement dynamics. Across such cases, simulating scenarios helps anticipate interventions before committing resources or execution, enabling more robust decision-making.

To address data quality challenges (13.43%). Data quality significantly impacts the reliability of analytical models. What-if analyses was conducted to tackle data quality issues, such as missing or incomplete data and uncertainty in the data by enabling users to simulate the effects of missing or noisy data and analyze how uncertainty in input variables influences model predictions.

Examples. Models often rely on ground truth labels for robust predictions, which are infeasible in real-world domains like climate simulation. Biswas et al. [7] addressed this by simulating a few scenarios to study the "sensitivity" of climate parameters on precipitation. Similarly, Zhang et al. [112] used "what-if analysis" to explore different promotion strategies under partially observed market conditions, simulating scenarios such as varying customer engagement. And, Hagele et al. [38] applied "sensitivity analysis" to examine how input uncertainty, like noise in student scores, affects projections of student rankings.

To leverage human knowledge in computationally intractable problems (26.11%). Exploring all possible parameter combinations or running thousands of simulations is often impractical due to resource and time constraints [86, 114]. What-if analysis was conducted to address these challenges, allowing users to focus on selected scenarios rather than exhaustively simulating all possibilities. By integrating human intuition and domain knowledge, these methods narrow the space of simulations, enabling more targeted exploration while significantly reducing computational overhead.

Examples. In high-dimensional anomaly detection, evaluating the impact of different algorithms and feature subspaces is challenging due to the large number of possible combinations and the sensitivity of results to parameter settings. Xu et al. [106] address this issue by enabling "what-if" analysis to systematically vary algorithm choices and parameters, helping users focus on promising configurations without exhaustively testing all possibilities. Likewise, in policy modeling, refining parameters, modifying interventions, and adjusting scenarios is an iterative, labor-intensive, and programming-heavy [85]. What-if analyses addressed such problems by automating "parameter exploration" [57], enhancing debugging through detection of dissimilar parameter setting scenarios [27], and performing "reverse mapping" to identify parameters for target outputs [79].

To mitigate cognitive biases (5.97%). Human biases frequently influence decision-making, potentially leading to skewed outcomes. In few works, what-if analysis was employed to identify, mitigate, or better understand these biases, providing insights into how different assumptions or data changes affect decisions.

Examples. Wang et al. [93] demonstrated how "exploring multiple hypotheses" helped mitigate healthcare experts' anchoring bias (on initial symptoms) and consider alternative diagnoses. Additionally, other works demonstrated observing many "counterfactual" scenarios to uncover audience biases, such as misinterpreting correlation as causation [9, 91] or drawing erroneous conclusions about feature-outcome relationships [45]. Further, integrating "constraints" when exploring counterfactuals, such as allowing realistic or domain-specific changes in features only, ensures them to be plausible and avoid leading to biased conclusions [10]. Further, "sensitivity analysis" also aided in addressing algorithmic biases by exposing how sensitive automated ranking and recommendation algorithms are to data perturbations [1, 104].

To incorporate user preferences and constraints (11.19%). Another motivation identified was to ensure that model predictions align with user needs and constraints. In many applications, users' specific needs, goals, and constraints shape how they interact with and derive value from predictions. What-if analysis can be a valuable tool for personalization, adapting the recommendations to such needs.

Examples. Wang et al. [98] developed an interface that iteratively generates "algorithmic recourse plans" of suggested changes to input variable values that would result in loan approval. It allows end-users to customize plans by specifying which features to modify and defining acceptable ranges for them, thereby aligning the recommendations with user preferences and constraints. Similarly, Gathani et al. [30] perform "constrained analysis" to help business users determine the best marketing spend across different channels while adhering to budget limitations. Lee et al. [52] recommend optimal sleep schedules, tailoring outputs based on users' individual routines and constraints.

3.2 What?

Many types of analyses, including but not limited to hypothesis testing, model evaluation, what-if analysis, and error analysis, require two components: a **`dataset`** and a predictive **`model`**. The ***what*** question describes the specific dataset and model `subcomponents` that distinguish what-if analysis from the other types of analysis.

B.1. **`Dataset`**

We found that what-if analysis has been applied to various types of datasets: multivariate (68%; 12% geo-related/spatial and 2% time series/temporal), image (10%), textual data (9%), audio (1%), and no datasets (12%), where the analysis was discussed theoretically or empirical data was not explicitly mentioned or explained. We define the following dataset `subcomponents` across these types:

- `instances`: fundamental units of data (e.g., rows in tabular datasets, individual images in image datasets, or entire documents or prompts in textual datasets)
- `features`: properties associated with each instance, varying by dataset type. For example, Wexler et al.'s [100] conducted what-if analysis on a tabular dataset with quantitative (e.g., *Capital Gain*), categorical (e.g., *Gender*), and ordinal (e.g., *Capital Gain*) features. Similarly, Park et al. [67] perform what-if analysis on images features like pixel intensities, color distributions, and texture patterns. In textual datasets, features range from words or sentences [55] to textual embeddings [2].
- `values`: specific instantiations of features For example, 'Male' or 'Female' for *Gender*, '0 to 5000' for *Capital Gain*, pixel intensities or color channel values (e.g., RGB values) in images, or words, n-grams, or embedding vectors in text.

B.2. **`Model`**

The **`model`** component captures the data feature relationships and predictions. It requires specifying `inputVariables` from the **`dataset`**, the `outputVariable` to predict, the supported `modelType`, and the `modelParameters`. For every what-if analysis performed, different models may be used.

B.2.1. Input Variables

These are dataset `features` that are considered inputs of the model and those who influence the `outputVariable`. They may already exist (`existingFeature`) in the dataset or be derived (`derivedFeature`) from existing features using simple arithmetic operations or complex statistical functions.

B.2.2. Output Variable

An `outputVariable` is a dataset feature acting as the dependent variable for the model. Like input variables, it can either already exist in the dataset (`existingFeature`) (e.g., *Income* output variable in Wexler et al [100] paper) or be derived (`derivedFeature`) from existing features (e.g., *Goodness* calculated by pairwise comparisons between all data points). Typically, all what-if analyses aim to achieve objectives related to a single output variable.

B.2.3. Model Type

This refers to the type of model, which can be characterized by several factors: the nature of the output variable (e.g., categorical vs. continuous), the underlying algorithmic family (e.g., tree-based, regression, neural networks), and the specific application context for which what-if analysis is employed. We broadly categorize the models based on whether the model is established or user-defined because it reflects how the model is obtained to use for what-if analysis:

(1) Established Models (`establishedModel`) (75.94%): Most works used well-established models like machine learning models (e.g., regression [22, 30, 47], GAMs [42, 98], neural networks [67, 95], SVMs [24, 36, 106]), deep learning models (e.g., CNNs [39, 54, 67], BERT [6, 56], sequence-to-sequence [82]), optimization models (e.g., SEIR [73, 85], Monte Carlo Simulations [8, 85]), dimensionality reduction models (e.g., PCA [14, 24, 65], MDA [65]), and domain-specific models (e.g., WRF climate model [7], GCAM forecasting model [60]).
(2) User-Defined Models (`userDefinedModel`) (15.04% specified by authors, 0.75% defined by end-users at runtime): A few works used specially defined models to allow tailoring to prior knowledge known about the relationships between variables. For example, in the marketing domain, Gathani et al. [32] model the *Impression* output variable as a linear combination of various input features; 2×*Paid Media Ads* + 1×*Website Visit* + 1×*Video Completion Rate*, while Gortler et al. [37] demonstrate a non-linear user-defined model by applying a scaling factor to the input covariance matrix, resulting in complex transformations of the projection directions.

Across the reviewed works, 35.34% of models were trained entirely by the authors [39, 93], 6.77% combined author-trained and user-defined models [95, 103], 24.81% used fine-tuned third-party pre-trained models [11, 77], and 9.77% employed algorithms newly developed by the authors [18, 92].

B.2.4. Model Parameters

These are internal variables learned during training that map inputs to outputs. Examples include `featureWeights` and `biases` in machine learning models, and `kernels`, `attentionWeights`, and `positionalEncoding` in deep learning models. For instance, Görtler et al. [37] varied a `scalingFactor` affecting covariance in PCA, while Zhang et al. [112] helped marketing experts rank influential promotions by learned `featureWeights`.

3.3 How?

The ***how*** perspective outlines the methods for conducting what-if analysis through two sets of operations: **`user_operations`**, initiated by users, and `system_operations`, executed by systems.

C.1. **`User Operations`**

To conduct what-if analysis, users interact with systems through a variety of operations that allow them to manipulate, explore, and reason about both the data and model behavior. Based on our analysis, we identified six common types of user operations. Among these, the first two—`scope` and `derive` (marked with *)—are general operations frequently observed in SQL querying and exploratory data analysis. Although not exclusive to what-if analysis, they often serve as essential operations that establish the necessary context for more focused what-if analysis.

C.1.1. Scope

The `scope` operation defines specific parts of the **`dataset`** as the focus of a what-if analysis. This operation can be applied to instances, features, or both. For example, Zhang et al. [112] allowed marketing analysts to narrow their what-if analysis by selecting specific `instances`, illustrating scoping at the instance level (`instanceSubset`). Scoping can also be feature-based. For instance, Gathani et al. [32] excluded economic indicators such as the *Consumer Confidence Index* and *Producer Price Index*, recognizing that business users cannot influence these variables. This represents a feature-level subset (`featureSubset`). The `scope` operation can also be applied beyond multivariate datasets. For instance, for textual datasets, Wu et al. [101] clustered unlabeled documents into semantic groups (e.g., holidays, date formats), illustrating instance-level scoping to examine model behavior within targeted subsets.

C.1.2. Derive

The `derive` operation creates new input or output features from existing ones, using simple arithmetic or complex transformations. For instance, Zgraaggen et al. [111] define the *Total* `outputVariable` by subtracting `inputVariables` *Expenses* from *Income*, while Ahn et al. [1], introduce the *Miscalibration* output variable via complex means like KL divergence measure to quantify algorithmic harms in a movie recommendation dataset.

C.1.3. Perturb

The `perturb` operation modifies the values of some components to examine their impact on other components. We identified two primary variations: perturbing `inputVariables` values and learn its effects on the `outputVariable`, and perturbing `modelParameters` to study how changes in the model's internal parameters affect its predictions or behavior. An example of the former is, Wang et al. [92] examining how reducing *Price* (`inputVariable`) by 10% would influence *Sales* (`outputVariable`), while of the later is Liu et al. [55] changing the *attention* (`modelParameter`) in natural-language-based convolutional neural networks to evaluate its impact on sentence-level inference tasks.

C.1.4. Constrain

The `constrain` user operation places restrictions on certain components and can be applied in two ways:

(1) Constraining **`model`** Function: Users can impose constraints on the **`model`** function to restrict the mapping between `inputVariables` and `outputVariable`. Examples include monotonicity constraints (e.g., ensuring that *Sales* should not continue to increase linearly after a threshold as *Unit Price* rises [31]), or fairness constraints that that exclude sensitive attributes such as *Gender* [5]. These operate at the level of model parameters and functional relationships.
(2) Constraining Variables: Users can directly limit the values or categories of specific `inputVariables` or `outputVariables`. For example, quantitative variables can be restricted between lower and upper bounds (e.g., constraining *Blood Sugar* between '4.7 and 6.3' [5]), while categorical variables may be fixed to specific values (e.g., fixing *Gender* to 'Female' [36]). Constraints can also express objectives such as `minimize`, `maximize`, or `closeTo(referenceDataPoint)`, for example, maximizing *Sales* [32], minimizing *Deal Closing Rate* [30], or ensuring prediction remain close to a reference point [86, 97, 98].

In summary, the `constrain` operation requires specifying both the *target* (a `modelParameter`, `inputVariable`, or `outputVariable`) and the *form* of `constrainValue` (`lowerBound`, `upperBound`, `categoryBounds`, or objective-oriented constraints like `minimize`, `maximize`, or `closeTo(referenceDataPoint)`).

C.1.5. Reweight

The `reweight` user operation modifies the relative importance expressed as weights of specific `inputVariables` within a model, guiding model predictions towards a desired direction. The reweighting can stem from users' domain knowledge or prior experiences. For example, Pajer et al. [66] reweighted `featureWeights` like *Price*, *Mileage*, and *Guarantee Period* in a car ranking model to observe their impact. It involves two key subcomponents: the `inputVariable` whose weight is to be adjusted, and the corresponding `featureWeight` assigned to it, allowing users to align model behavior with real-world needs or expectations.

C.1.6. Compare

The `compare` operation lets users actively select and juxtapose the outcomes of multiple what-if analyses. Unlike atomic operations such as `perturb` or `constrain`, which modify variables or models directly, `compare` operates on the results of executed scenarios. In practice, the user specifies which scenarios to include in the comparison–such as different subsets of the dataset (via `scope`), alternative perturbations of variables (via `perturb`), or varied model or variable constraints (via `constrain`)–and the system aligns their outputs along common axes (e.g., predicted outcomes or performance metrics).

Comparison is crucial for evaluating trade-offs and relative impacts across scenarios. For example, Bhattacharya et al. [5] contrasted treatment recommendations across patients, Evirgen et al. [25] examined prompt variations in text-to-image generation, and Park et al. [67] assessed classifier robustness under parameters like *Rotation* and *Blur*. Thus, while `compare` operates at a higher abstraction level, it remains a key operation within our framework because it formalizes the user's action of selecting and contrasting scenarios to reason across what-if outcomes.

C.2. **`System Operations`**

In response to **`user_operations`**, systems perform internal operations to carry out the requested transformations or updates. We identified six types, with the first two—`transform` and `expand`—being general-purpose operations common in broader data processing and analysis workflows. While not unique to what-if analysis, they are included in our framework to support more targeted analyses.

C.2.1. Transform

In response to the `scope` user operation, the system automatically executes the corresponding `transform` system operation to refine the **`dataset`** based on the defined subset. This operation ensures that downstream analyses only act on the relevant portion of the data. This process mirrors the data transformation techniques like "filter" and "slice and dice" techniques commonly used in exploratory data analysis. For example, Dingen et al. [22] scoped their analysis to 'Secondary' *School Students* filtered by the 'Math' *Subject* in the 'Portugal' *City*, producing a specific subset that informed where to perform the subsequent what-if analysis.

C.2.2. Expand

The `expand` system operation extends the **`dataset`** by adding newly derived features, making them available for downstream what-if analysis. It is automatically triggered by the `derive` user operation, which enables users to define new features of interest. By reflecting these transformations in the data, the system supports more tailored what-if analyses.

C.2.3. Rerun

The `rerun` system operation re-executes the model's inference or evaluation pipeline to generate updated predictions based on the current state of the system. It is automatically triggered by user operations that modify the **`dataset`** (e.g., `scope`, `derive`, `perturb`) or alter the model configuration, such as updates to `modelParameters` through operations like `reweight` or `constrain` on `inputVariables`. By reapplying the model with these updated inputs, the system generates revised predictions to support interactive what-if analysis.

C.2.4. Optimize

The `optimize` system operation is performed to achieve the desired objective on the `outputVariable`. It computes `inputVariables` values required to achieve the desired objective based on the `outputVariable`, based on the learned function of the **`model`**. For instance, Cheng et al.[15] identified required *Test Scores* and *GPA* for getting university admission.

In our reviewed works, the `optimize` system operation is often invoked following the `constrain` user operation. This is because once users specify bounds on permissible `inputVariables`, desired outcomes for the `outputVariable`, or objectives (`maximize`, `minimize`, or `closeTo(referenceDataPoint)`), the system searches the feasible space to find optimal `inputVariable` configurations. Mainly three variations were observed:

(1) `maximize` the `outputVariable`: This aimed to attain the maximum value of output variable, such as for *Sales* [30, 32], *Income* [100], or *Retention Rate* [30].
(2) `minimize` the `outputVariable`: This aimed to attain the minimum output variable value, such as for *Risk of Diabetes* [5] or *Number of Infections* [85].
(3) attain a `target` `outputVariable` or `closeTo` a `target` `outputVariable`: Adjust `inputVariables` to make the `outputVariable` reach a desired `target` outcome or approximate a desired target or reference point. For example, keeping *Precipitation* estimates close to baseline [41]. A special case is *counterfactuals*, where the system finds minimal input changes to yield a different outcome [5, 40, 77].

C.2.5. Retrain

The `retrain` system operation updates the **`model`** when user operations alter either the **`dataset`** or the **`model`** itself. Retraining can occur in two ways based on what component is modified. In one case, the same model is retrained because of component adjustments. For example, user operations like `reweight`, which modify `featureWeights`, change the relative importance of input variables, requiring the model to relearn its internal representations. Pajer et al. [66] illustrated this in a car ranking system, where altering the weights of features prompted model retraining to reflect new ranking priorities. Similarly, operations like `scope` and `derive`, which filter the dataset to a subset or introduce new features, respectively, alter the data distribution or structure, necessitating retraining to maintain consistency and validity in predictions. In another case, the model itself is changed. For instance, Wexler et al. [100] allow to compare what-if scenarios across different model types, such as linear classifiers and neural networks. Therefore, changing the underlying user operations or model architecture triggers retraining to ensure that predictions remain accurate.

C.2.6. Generate

The `generate` system operation produces a comprehensive set of what-if analysis scenarios each consisting of a configuration of `inputVariables` and its corresponding `outputVariable` predictions computed using the underlying **`model`**. This operation enables users to explore the many possible inputs and their predicted outcomes to identify interesting or actionable scenarios. The generated scenarios share the analytic context with from variations across specific components. For example, scenarios may apply the same `perturb` operation across different data segments via `scope` (e.g., geographic regions [30]), or generate past-case analogs (e.g., credit card defaulters) under varying constraints that must be satisfied [92].

3.4 Types of What-If Analysis

From our review of the components and how they were combined to perform what-if analyses, we identified three primary types of what-if analyses: (1) **`SENSITIVITY`**, (2) **`GOAL SEEK`**, and (3) **`IMPORTANCE`**. In addition, we observed a broader fourth type, **`SCENARIO COMPARISON`**, which often integrates aspects of the others. Figure 4 illustrates how each type is

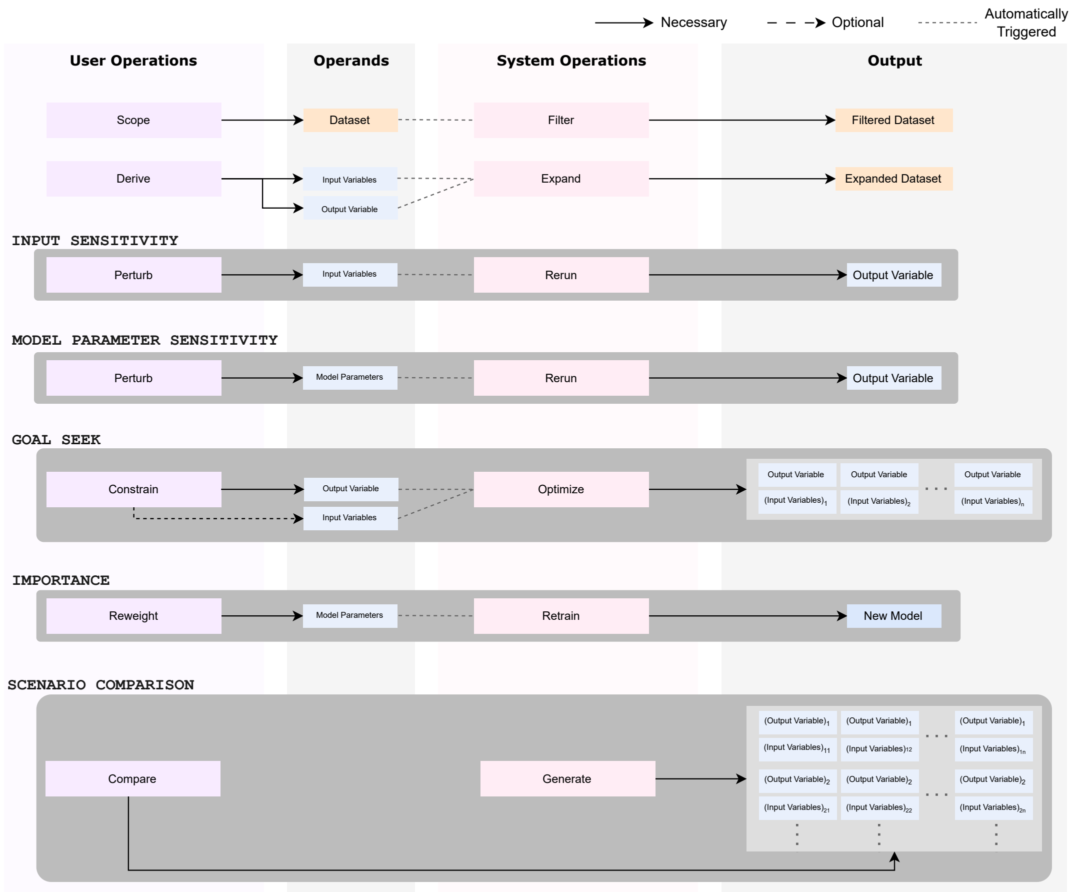


Fig. 4. Combining different `components` of what-if analysis to form its different **TYPES**.

operationalized: the user operations applied to different ***what*** components, the corresponding system operations that are automatically triggered, and the resulting analytical outputs.

Each type aligns with core user operations: **SENSITIVITY** via `perturb`, **GOAL SEEK** via `optimize`, and **IMPORTANCE** via `reweight`. We define these types below, formalize them as: `what-if analysis type function(parameterized by components)` $\rightarrow$ `output`, where `:=` denotes the definition of each component. We also provide examples from our review and highlight representative visualizations used for both conducting each what-if analysis and for observing its predictions. Extra details can be found in supplementary materials [1].

D.1. SENSITIVITY

SENSITIVITY what-if analysis–often referred to as various terminologies such as "perturbation-driven exploration" [55], "parameter sensitivity" [7, 73], "forward-mapping" [14, 81], "what-if analysis" [21, 41, 109], among others–is a systematic approach to understand how altering specific input variables or model parameters influence the output variable or model predictions.

We categorize **SENSITIVITY** what-if analysis into two primary types based on what is being altered:

(1) **INPUT SENSITIVITY** (49.25%): Varying values of `inputVariables` to examine the resulting effects on the `outputVariable`. For example, Victor et al. [89] explored how adjusting production parameters such as *Spinning Speed*, *Number of Spin Positions per Meter*, and *Belt Speed* influences the *Product Quality* of nonwoven materials.

(2) **MODEL PARAMETER SENSITIVITY** (18.66%): Altering **`model`** parameter values to examine their effect on the model predictions. For example, Liu et al. [55] perturbed the internal states of a neural network-based machine learning model by adjusting the `attention` weights assigned to individual words or word pairs. This allowed them to analyze the impact on the model's classification of a sentence as *entailment*, *contradiction*, or *neutral*.

Before conducting any type of **SENSITIVITY** analysis, user operations like `scope`, `derive`, or both may be optionally applied to narrow analysis to a desired subset of dataset or have new features be added to the analysis respectively. But, the main operation enabling **SENSITIVITY** analysis is the `perturb` operation, which is used to specify the alterations. Based on the perturbation type (`perturbType`) described in the earlier section, two approaches of performing the **SENSITIVITY** what-if analysis can be taken:

(1) over Individual Points, when `perturbType = pointValue`; a single adjustment is applied per input variable or model parameter. (2) over Continuous Ranges, when `perturbType = rangeValue`; adjustments over a range of values are made to input variables or model parameters to learn the prediction trend of the output variable.

In practice, users may also apply constraints via the `constrain` operation (e.g., fixing certain variable values while varying others, or limiting changes to feasible ranges) to ensure realistic conditions for the analysis. Every time the `perturb` operation is applied, the system automatically executes the `rerun` operation to re-evaluate model predictions based on the modified inputs.

```
SENSITIVITY(α, β, γ, δ) → outputVariable
α: dataset,
β: model,
γ: user_operations := {[scope(dataset)]* | [derive(newFeature)]* | [constrain(inputVariables)]*}
& [perturb(inputVariable|modelParameter, perturbType)]+,
δ: system_operations := [rerun()]+
```

Visualization & Interaction. This type of analysis requires interactive perturbation of system components to examine output effects. In our review, the most common input interactions for the **SENSITIVITY** analysis type were *sliding a numeric parameter* (Fig. 5A [67]; see also [43, 51]), and *selecting categorical variables* (Fig. 5B [41] see also [30, 86]) (33%). Less frequent interactions include *editing input values directly in tables* (Fig. 5C [100]; see also [85, 105]) (5%), and *textual modifications* (Fig. 5D [55]; also [112]) (10%). We also observed research prototypes that allow *dragging data points in visualizations* (Fig. 5E [65]; see also [22, 111]) (15%), or *spatial manipulation of structures* (Fig. 5F [104]; also [80]) (5%). Although promising, these latter techniques have seen limited adoption, likely due to higher implementation effort.

In contrast, results from the **SENSITIVITY** analysis type were communicated through a wide range of visualizations, though some clear preferences emerged. Scatterplots were the most common (21%), reflecting most works discussing classification tasks when discussing relationships between input perturbations and output changes. Heatmaps (15%) were also widely used, especially when comparing sensitivities across multiple variables. Line charts (14%) and bar graphs (12%) provided straightforward ways to track trends and compare magnitudes.

Other representations appeared less frequently. Tables (8%) were often paired with charts to provide detailed numerical validation. Graphs and trees (9%) and maps (9%) supported more specialized hierarchical or spatial scenarios. Parallel

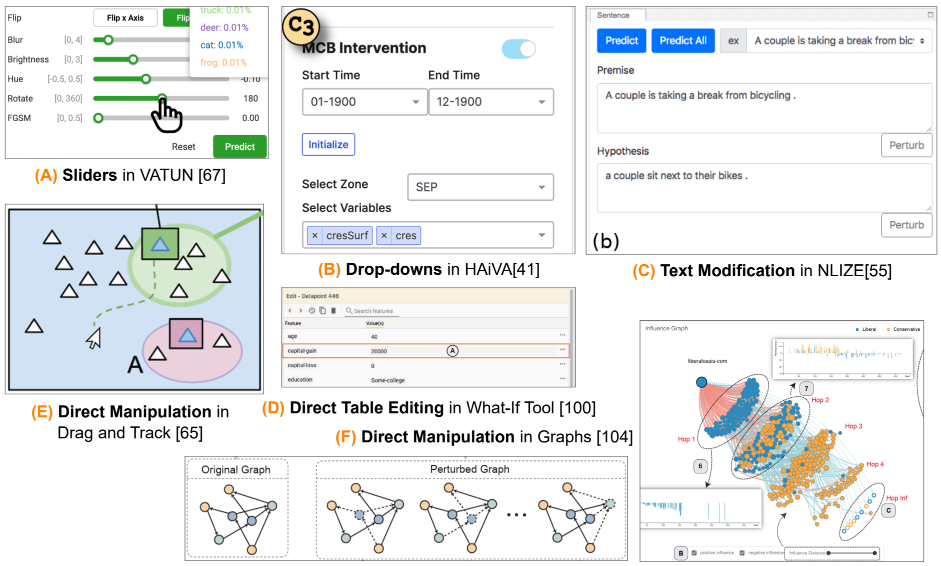


Fig. 5. Examples of input interactions for **SENSITIVITY** analysis type from reviewed literature.

coordinate plots (PCPs) (4%) were rare, despite their suitability for high-dimensional data, suggesting barriers to adoption. A small set of domain-specific visuals–including icicle plots, donut plots, flowcharts, and matrices (1–2% each)–were seldom used, likely due to their complexity. Textual annotations (5%) and images (4%) occasionally supplemented visual outputs, while clustering techniques (3%) and box plots (3%) appeared sporadically.

D.2. **GOAL SEEK**

In **GOAL SEEK** what-if analysis, the aim is to determine the combinations of `inputVariable` values that achieve the optimize the `outputVariable` according to the user-specified goal. We found 17.16% of reviewed works conducting this type of what-if analysis, often referred to by different terminologies, such as “inverse modeling” [63], “segmentation” [81], “backward mapping” [14, 81], “parameter setting” [57], all representing the same analysis.

The system employs the `optimize` operation to explore and evaluate multiple input variable configurations that meet the defined goal. Often, users further refine the search space using the `constrain` operation, which allows them to restrict variables within explicit bounds or categories, fix maximize or minimize objectives on them, or specify needs of being close to certain input values. Applying such user operation helps incorporate domain knowledge and external constraints, ensuring the predictions adhere to realistic or predefined conditions.

For example, Tariq et al. [85] conduct this analysis to seek the optimal policy involving input variables like *School Closure*, *Masks*, *Hospital Capacity*, etc. to minimize the *Number of Infections*. Similarly, Wang et al. [98] perform this analysis to determine input variables like *Credit Utilization*, *Home Ownership*, *Loan Amount*, among others to achieve the target outcome of *Loan* being ‘Approved’. Wang et al. further incorporate constraints or user needs; for instance, restricting the *Loan Amount* between acceptable ranges ‘25000’ to ‘40000’, ensuring realistic recommendations. The outputs of this analysis include both the optimized `outputVariable` and the corresponding solution paths of the optimized `inputVariable` values. For instance, adjustments in Wang et al.’s analysis include setting *Loan Amount* to ‘30000’, reducing *Credit Utilization* from '90.1' to '63.75', changing *Home Ownership* from 'Rent' to 'Mortgage', among other modifications.

```
GOAL SEEK(α, β, γ, δ) → (inputVariable:values & outputVariable)+
α: dataset,
β: model,
γ: user_operations := [scope(dataset)]* | [derive(newFeature)]* |
[constrain(inputVariable|outputVariable)]+,
δ: system_operations := [optimize()]+
```

Visualization & Interaction. The most common interfaces involved *sliding numeric parameters* (Fig. 6A [15] (50%); see also [45, 54, 79, 98]) and *selecting categorical conditions*, such as drop-down menus (Fig. 6B [5]; see also [30]), radio buttons [68, 101], or checkboxes [53, 61] (33%). Other forms included *entering target values in textboxes* (Fig. 6C [30]; see also [8, 18, 31, 111]) (16%) and *defining ranges through region selections* (Fig. 6D [113]; see also [3, 70, 103]) (8%). While these interactions allowed users to explicitly define constraints, some systems bypassed direct input altogether, instead predicting and presenting optimized scenarios automatically based on predefined or implicit objectives (Fig. 6E [99]; see also [6]) (6.3%).

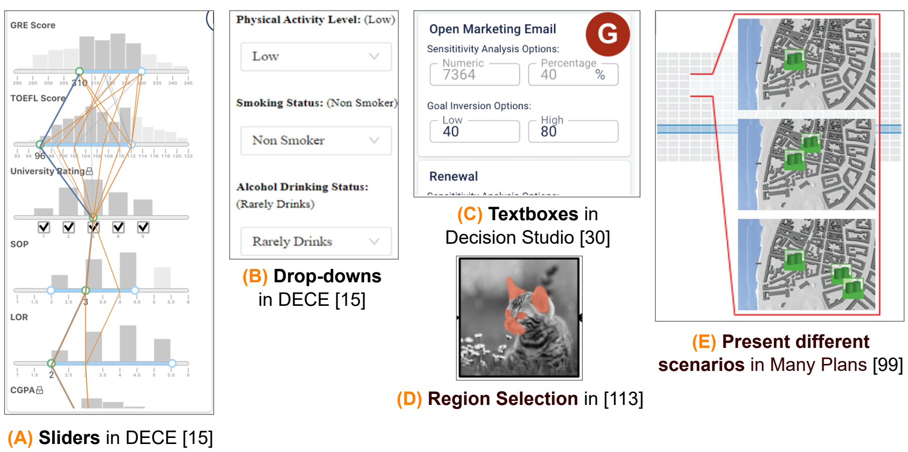
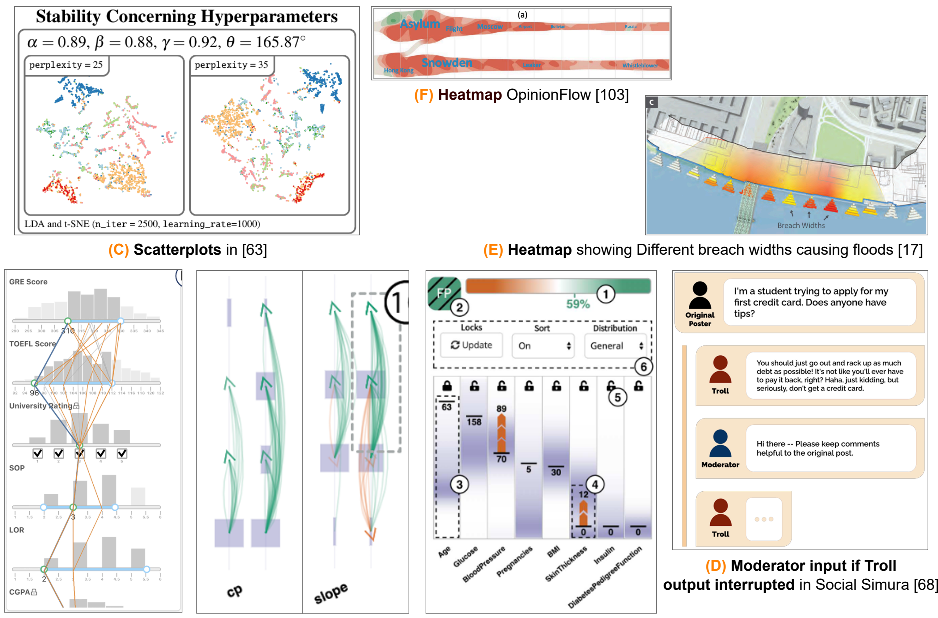


Fig. 6. Examples of input interactions for **GOAL SEEK** analysis type from reviewed literature.

The results of **GOAL SEEK** analysis typically included optimized input variable values along with their corresponding predicted output values. These were often conveyed through a combination of visual representations. For example, PCPs showing various input variable combinations or change from original values (Fig. 7A [15, 36]), annotations and heatmap showing change in input variable values (Fig. 7B [35], E [17], F [103]) (29%), line charts showing output value progression relative to selected inputs [54, 111] (20%) and scatterplot distribution shift for various model hyperparameters (Fig. 7C [2]; see also [18, 92]) (12.5%).

D.3. **IMPORTANCE**

The **IMPORTANCE** what-if analysis comprises 19.40% of reviewed works. This analysis is referred to under various terminologies, such as "influential factors" [105, 112], "feature-importance scores" [42, 45, 51], "driver importance analysis" [30, 32], "guided back-propagation" [54], among others, but all represent the same fundamental concept.



(F) **Heatmap** OpinionFlow [103]

(C) **Scatterplots** in [63]

(E) **Heatmap** showing Different breach widths causing floods [17]

(D) **Moderator input if Troll output interrupted** in Social Simura [68]

(A) **PCPs** in DECE [15] and Advice [36]

(B) **Annotations, heatmaps** in ViCE [35]

Fig. 7. Examples visuals used for showing results for **GOAL SEEK** analysis type from reviewed literature.

The purpose of this analysis is to understand the significance of `inputVariables` as a predictor for the `outputVariable`. Typically, this is achieved by computing `featureWeights` using techniques such as SHAP, LIME, probabilistic significance, or model coefficients. These feature weights rank input variables from most to least influential based on their contribution to the model's predictions. Most related works do not incorporate direct user interaction but instead focus on observing feature importance through scores [5, 30, 32, 42, 56] or rankings [25, 51, 66]. For example, Wang et al. [97] investigate which words in a prompt are most influential in improving predicted text's *Quality* and *Diversity* scores. Similarly, Zhang et al. [112] analyze the factors influencing *Sales Volume*, identifying that *Price* and *Competition* consistently have higher importance compared to *Promotion*, which has a lesser impact.

Additionally, users may actively adjust the significance of input variables via the `reweight` user operation, allowing users to prioritize specific input variables based on their preferences. For example, Pajer et al. [66] allowed users to assign custom weightage to input variables like *Price*, *Mileage*, and *Age* when ranking cars with goals to minimize *Price* and maximizing the *Guarantee Period*.

The `scope` and `derive` user operations can optionally be applied in this type of analysis to restrict the input variables considered as well as add desired features in the analysis. This operation allows users to focus on the significance of specific input variables concerning the output variable according to users' needs. For instance, Gathani et al. [32] rank the importance of investments made on advertising channels to determine which channels to invest in to increase the *Sales*. After these user operations are performed, the system automatically triggers the `retrain` operation to update the model based on the changes.

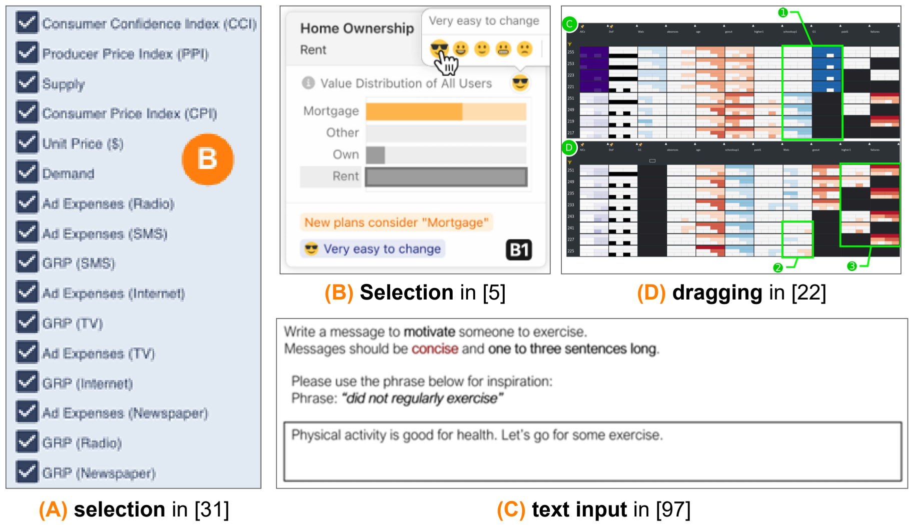


Fig. 8. Examples of input interactions for **IMPORTANCE** analysis type from reviewed literature.

```
IMPORTANCE(α, β, γ, δ) → refinedModel
α: dataset,
β: model := featureWeights,
γ: user_operations := [scope(dataset)]* | [derive(newFeature)]* | [reweight(inputVariables)]+,
δ: system_operations := [retrain()]+
```

Visualization & Interaction. Most reviewed works in **IMPORTANCE** analysis required minimal user interaction, with most systems automatically computing feature importance scores based on model internals. When user input was present, it typically involved selecting relevant input variables or reweighting them to reflect user priorities. Across the reviewed works, these input interactions were employed via selection (Fig. 8A [31] and B [98]) (38%), text inputs (Fig. 8C [97]) (27%), and sliders (15%) [5, 98] being the most common. Less frequent input interactions included direct *dragging* on visualizations (Fig. 8D [22]; see also [3]).

Output visualizations in **IMPORTANCE** analyses primarily used bar charts (Fig. 9A [30, 31]) (21%) and heatmaps (Fig. 9C [22]; see also [56]) (15%) to display feature importance scores. Other outputs included ordered list of text outputs (Fig. 9B [42]) (12%) to present ranked variable lists or numerical importance values. In textual datasets, importance was illustrated using lesser observed illustrations like word clouds (Fig. 9E [56]) and saturation encoding (Fig. 9D [25]).

D.4. SCENARIO COMPARISON

The **SCENARIO COMPARISON** analysis type, present in 19.40% of the reviewed works, reflects a broad and compound type that spans across the other what-if analyses. It centers on the `generate` system operation, which automatically produces a diverse set of plausible `inputVariable` configurations and their associated `outputVariable` predictions without requiring users to specify goals or manually manipulate variables. This allows users to explore a wide spectrum of possible scenarios, supporting open-ended investigation into model behavior and surfacing unexpected or interesting relationships between inputs and outputs.

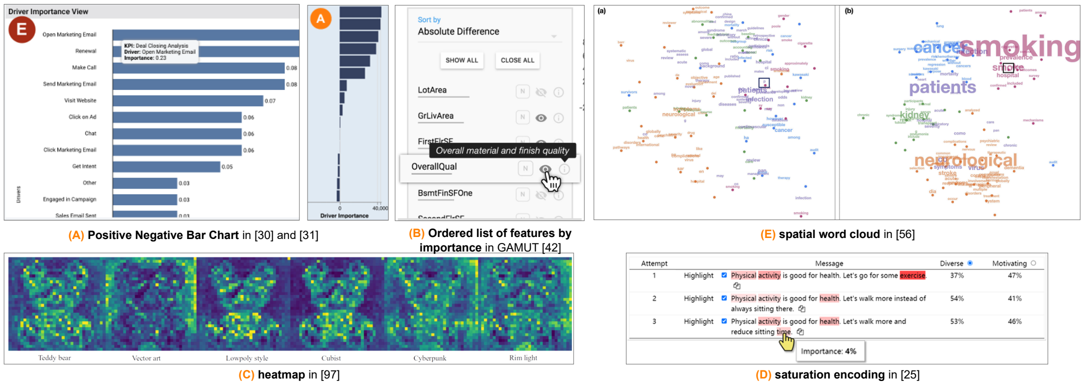


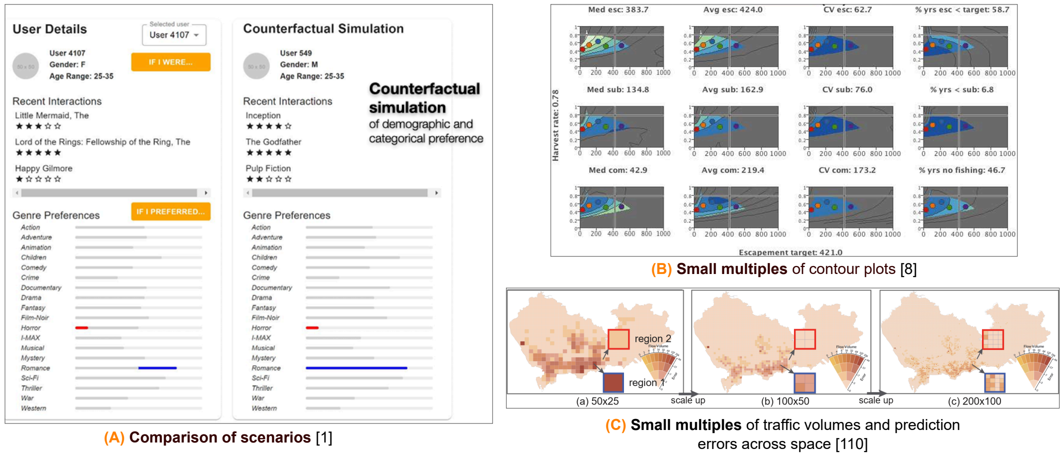
Fig. 9. Examples of output visuals for **IMPORTANCE** analysis type from reviewed literature.

While it may share similar components as **GOAL SEEK** or **SENSITIVITY** analyses types, it differs in intent and interaction. In **GOAL SEEK**, users specify a goal of the `outputVariable` or a desired output of it, and optimization is conducted to find combinations of `inputVariable` values to achieve it. In **SENSITIVITY**, users perturb `inputVariables` to observe changes in `outputVariable`. In contrast, **SCENARIO COMPARISON** does not take any explicit objective or input manipulation but automatically generates and presents the users with many scenarios belonging to the same analysis type. In response, to make sense of the many generated scenarios, users often engage in the `compare` operation—noting model outputs one scenario at a time or juxtaposing scenarios to examine variations across different component interventions. For example, Browne et al. [11] generate perturbed images to explore visual resistance strategies against *Corporate Surveillance*. Wang et al. [92] take a similar approach by providing multiple input scenarios with similar starting conditions but different predicted outputs. Some works also provide comparison features to distinguish these scenarios from one another [78, 113].

```
SCENARIO COMPARISON(α, β, γ, δ) → (inputVariable:values & outputVariable)+
α: dataset,
β: model,
γ: user_operations := {[scope(dataset)]* | [derive(newFeature)]*} & [compare(inputVariable:values &
outputVariable)+],
δ: system_operations := [generate(inputVariable:values & outputVariable)]+
```

Visualization & Interaction. Similar to **SENSITIVITY** and **GOAL SEEK** analysis, a few of the **SCENARIO COMPARISON** analysis employed sliders [48, 77, 111], selection tools such as drop-downs [10] and checkboxes [61], and textboxes [13, 77, 111] to filter down to different scenarios. But some reviewed works did not require any user input; instead, they automatically generated alternative scenarios for exploration and comparison (Fig. 10A [1]; see also [54, 78, 110]).

The visual representation of this type closely resembled those used in **GOAL SEEK** analysis. Common approaches included bar charts [13, 77, 96], line charts [10, 54, 92], histograms [54, 61], and scatterplots [48, 92, 110] to illustrate the range of possible input values in *small multiples* (Fig.10B [8] and C [110]). Additionally, some works present raw tables listing multiple generated scenarios alongside their predicted output values [44, 54, 111].



(A) **Comparison of scenarios** [1]

(B) **Small multiples** of contour plots [8]

(C) **Small multiples** of traffic volumes and prediction errors across space [110]

Fig. 10. Examples of output visuals for **COMPARE SCENARIOS** analysis type from reviewed literature.

4 Case Studies

To demonstrate the practical utility of our framework, we present two case studies that illustrate how complex what-if analyses can be deconstructed into their fundamental components and categorized into the specific types of what-if analyses we find. In the first case study, we examine how a commonly discussed analysis of "counterfactuals" could be understood as either **GOAL SEEK** or **SCENARIO COMPARISON** analyses, depending on users' motivation and operation. The second case study focuses on "parameter space exploration", especially in dimensionality reduction tasks which can align as **SENSITIVITY** or **GOAL SEEK** analyses types.

4.1 Counterfactuals Across Different WIA Types

Many reviewed works focus on providing "counterfactuals", but the terminology has been used in different ways across domains. In machine learning interpretability, counterfactuals are typically defined as the *minimal change in data that flips the model's prediction* [15], with the goal to provide actionable recourse for users or to get transparency in decision boundaries of a model. In the context of causal inference, the lens shifts to potential outcome under different intervention conditions than the ones actually observed, for example *what needs to change (i.e., intervention) for the alternative outcome (i.e., potential outcome) to happen* [93] or *how to change inputs (i.e., intervention) to get Q instead of P (i.e., contrasting potential vs observed outcome)?* [97]. Further, in the visualization context, counterfactuals is framed more loosely as the *changes to input features that would lead to an expected model prediction* [98] or *instances similar to a given subset, but yielding different predictions* [45].

Despite these differences, all the variations fundamentally share the same analytical goal: to identify what input values would lead to a desired or alternative outcome. According to the PRAXA framework, all of these variations map to the **GOAL SEEK** type of what-if analysis, but they differ in the components emphasized, the kinds of constraints applied, and the goals users pursue. We illustrate this across three representative cases.

Counterfactuals in ML Interpretability. Cheng et al. [15] developed a system that helps non-admitted students explore how profile features such as *GRE score*, *TOEFL score*, and *CGPA* could be minimally adjusted to flip the prediction from *Rejected* to *Admitted*.

- `system_operation`: `optimize` searches for the smallest changes to change the model's output from 'Rejected' to 'Accepted'.
- `user_operation`: Students apply `constrain` to keep modifications realistic, e.g., bounding *GRE* between '320–340' or *University Rating* '$\leq$ 4.5'.

In this setting, counterfactuals operate as **GOAL SEEK** by revealing the smallest feasible adjustments to features while staying within realistic bounds.

Counterfactuals for causal inference. Wang et al. [97] illustrates counterfactual suggestions as asking: what would the message's *quality* and *diversity* be when *alternative words* are proposed (e.g., replacing terms with semantically related ones). Here, the counterfactual is not about flipping or achieving any target outcome but about exploring potential outcomes under different interventions at the word level. This maps to **GOAL SEEK** what-if analysis type but with the following components:

- `system_operation`: The `optimize` operation is used to propose word edits that move a message toward higher-quality or more diverse predictions.
- `user_operation`: The `constraint` operation is implicitly applied by choosing which counterfactual word substitutions are feasible or contextually appropriate.

Here, counterfactuals take the form of **GOAL SEEK** by examining how outcomes shift under alternative interventions, contrasting potential with observed results.

Counterfactuals in visualization context. Wang et al. [98] generate counterfactual recourse plans for loan applicants by exploring alternative values of features such as *FICO Score*, *Credit Utilization*, or *Annual Income*. The output variable is whether *Loan Approved* or *Not Approved*.

- `system_operation`: `optimize` proposes feasible feature adjustments that achieve *Loan Approved*.
- `user_operation`: Users optionally apply `constrain`, e.g., fixing *Payment Period* at '36 months' or bounding *Loan Amount* between '$25,000 and 40,000'.

In this case, counterfactuals align with **GOAL SEEK** through the generation of many diverse, actionable recourse options rather than a single minimal change.

In some cases, especially when generating a range of plausible alternatives, counterfactual analyses may also resemble **SCENARIO COMPARISON**, where multiple input variable configurations or decision paths that satisfy different outputs are surfaced side by side without any user interaction or preference of a single optimization objective or constraint. For instance, Gomez et al.[36] explore diverse paths that either lead to or avoid heart disease, enabling users to contrast across different health trajectories.

Thus, while many works frame their contributions as "counterfactuals", in practice they operationally map with the **GOAL SEEK** what-if analysis type and also often extend to higher-level analyses like **SCENARIO COMPARISON**. By deconstructing these analyses into PRAXA's components, we can better capture their intent and type.

4.2 Extending What-If Analysis to Dimensionality Reduction and Parameter Space Exploration

Our review also revealed that the components and types of what-if analysis defined in PRAXA extend beyond traditional decision-support tasks into more complex domains such as dimensionality reduction, simulation ensembles, and parameter space exploration. These tasks often require reasoning across multiple levels of abstraction–e.g., between high- and low-dimensional data spaces or across ensembles of parameter configurations–making them interesting to examine how PRAXA clarifies overloaded terminology and unifies disparate analytic strategies.

A compelling analysis terminology was the "inverse projection" technique adopted by Espadoto et al. [24], which uses deep learning to map 2D projected points back into high-dimensional representations. Conceptually, this mirrors the **GOAL SEEK** analysis type, but with inputs and outputs defined not as individual data features or instance-level values but as entire data spaces. Within PRAXA, this translates to the `optimize` system operation to retrieve the high-dimensional data points corresponding to the low-dimensional data points. Despite the change in granularity of the data space, the underlying analytical goal of identifying input configurations that meet a specific output remains consistent with goal-seeking tasks. Other dimensionality reduction works also align similarly align. For instance, Torsney et al. [86], Kumpf et al. [48], and Molchanov et al. [62] all study interactive "parameter tuning" to steer outputs toward desired results. These can be precisely described as **GOAL SEEK** analyses combining `constrain` and `optimize` operations.

Complementary to these **GOAL SEEK** or "backward mapping analyses" are "forward mapping" analyses, exemplified in the work by Gortler et al. [37], who alter model parameters to explore the variability in low-dimensional projections produced by PCA. This aligns with the **MODEL PARAMETER SENSITIVITY** analysis type within PRAXA, where users perform controlled variations of input (e.g., projection parameters) to observe their impact on model outputs. Such a pattern involves a `perturb` user operation and often pushes users to `compare` outcome differences across scenarios. Splechtna et al. [81] extend this further by implementing both "forward" and "backward" mapping strategies across multiple simulation ensembles. In "forward mapping", they vary parameters to understand the shapes and boundaries of resulting output regions, resembling **SENSITIVITY** analysis. In contrast, their "backward mapping" approach attempts to find which combinations of control parameters lead to specific output patterns, aligning with **GOAL SEEK**.

The same broad what-if analysis types underlie within complex domains such as dimensionality reduction, ensemble simulation, and parameter space exploration–where users reason across multiple levels of granularity and abstraction. What may appear as distinct paradigms (e.g., inverse projection, parameter tuning, or forward vs. backward mapping) can be clearly described using PRAXA's components and types, revealing them as variations of **SENSITIVITY** or **GOAL SEEK**. In this way, PRAXA not only disambiguates overloaded terms but also unifies seemingly unrelated analyses under a common framework. More importantly, it highlights underexplored opportunities–such as very few mentioned and applications of richer uses like **COMPARE SCENARIOS** and almost no discussions around chaining analyses (e.g., plugging outputs of one analyses into another for continued analyses)–that could lead to novel future what-if analyses.

5 Limitations of the PRAXA Framework

While our framework provides a structured approach to understanding what-if analysis by defining its key components and demonstrating how combining them leads to different types of analysis, it is important to acknowledge certain limitations.

First, our framework is derived from an extensive review of existing literature, aiming to consolidate a broad spectrum of terminologies related to what-if analysis. However, we do not claim it to be exhaustive. As the field continues to evolve–particularly with advances in generative AI, interactive LLM agents, and multimodal reasoning–new analytical approaches may emerge that are not currently captured within our framework. Thus, it should be considered a dynamic framework, open to refinement and expansion as future research progresses.

Second, while our literature review was extensive, our framework has not yet been formally applied in the development of systems or tools designed to perform such analyses. Such applications would enhance the reproducibility and practical validation of our framework. Future work could explore implementing our framework as a foundation for developing what-if analysis features within systems and tools to assess its usability, completeness, and reproducibility.

Third, the boundaries between what-if types can be fuzzy and open to interpretation. For instance, **SCENARIO COMPARISON** is a higher-level analysis that compares other analyses like **SENSITIVITY** or **GOAL SEEK**. Gomez et al. [36]

developed a tool that explores all possible paths of demographic and health variables leading to heart disease, which fits **SCENARIO COMPARISON**. Yet, when users filter results to focus on cases where "disease does not occur", the analysis shifts toward **GOAL SEEK**, as it applies constraints and optimization to find inputs achieving a target outcome. Likewise, if users adjust a single variable (e.g., *cholesterol* levels) to see how predictions change, the task resembles **SENSITIVITY**. In this way, **SCENARIO COMPARISON** can encompass multiple finer-grained analyses, and interpretation depends on the operations performed and the user's intent.

Further, the same high-level analysis goal can often be achieved through different sequences of user operations. For instance, consider the goal of understanding how *cholesterol levels* affect *heart disease risk*. A user might (1) directly vary cholesterol values across a range (`perturb`) to observe how predictions change, (2) run a **GOAL SEEK** analysis and then filter the resulting scenarios to patients with cholesterol above 200 (`scope`), or (3) set cholesterol '= 200' as a hard requirement before optimization begins (`constrain`), prompting the model to rerun under this restriction. Although the end goal is the same, the operations differ in order and intent, suggesting different analysis types. While our framework provides language to describe these alternatives, interpreting them still requires human judgment or system-level guidance.

6 Relation to Other Framework Papers

Several existing frameworks partially overlap with our work but differ in focus and scope. For example, La et al. [50] survey visual analytics for explainable deep learning, recognizing perturbation and counterfactual techniques. We extend beyond this by systematically identifying components of these what-if analysis techniques and generalizing across a wider range of analysis types.

Von et al. [90] review visual analytics for optimizing structured pipelines (i.e., sequences of algorithmic operations that transform input into output) through eight application case studies. In contrast, our framework is derived from 141 papers and emphasizes execution and implementation of what-if analyses by decomposing them into `dataset`, `model`, and `outputVariable` components.

Further, Piccolotto et al. [69] review visual parameter space exploration, focusing on how model parameters relate to outputs in spatial and temporal contexts. While overlapping in sensitivity and optimization, their framework emphasizes high-level workflow actions (e.g., manual or automatic parameter setting). In contrast, our framework is domain-agnostic and broader: we position parameter space analysis as as a what-if analysis-related analysis and instead break down analysis into constituent components, enabling systematic characterization of different what-if types.

Sedlmair et al. [75] outline high-level data flows, navigation strategies, and six tasks in parameter space analysis. However, their framework does not fully address the specifics of how these flows and strategies relate to different analysis tasks. Our framework extends theirs by mapping these abstractions to specific what-if operations and types. For example, *optimization* and model output *fitting* correspond to our **GOAL SEEK** type, while *sensitivity* aligns with our **SENSITIVITY** type. Other tasks such as *outliers* and *uncertainty* are treated as motivations, and *partitioning* relates to our **SCENARIO COMPARISON**. Thus, whereas Sedlmair et al. emphasize task-level navigation, we provide a finer-grained grammar that explains how different components combine to support diverse what-if analyses.

In summary, existing frameworks typically address specific application domains or emphasize high-level abstractions, whereas our framework provides a fine-grained, component-level view of what-if analysis that is both systematic and extensible across domains.

7 Challenges and Opportunities for What-if Analysis

In addition to identifying key `components` and categorizing distinct **`TYPES`** of what-if analysis from reviewing prior work, we also distill recurring challenges discussed in the literature around conducting and interpreting such analyses. Here we discuss them and identify open research opportunities.

Deciding which analysis to execute and explore. Before users begin performing what-if analysis, they face the critical challenge of deciding what to explore. This requires a complex mix of analytical foresight, domain understanding, and cognitive tracking–all of which can present substantial barriers.

Selecting an appropriate analysis configuration often *required substantial domain knowledge* to configure models, choose appropriate input variables, and define meaningful constraints. Many systems expect users to make such decisions and decide on what to explore on their own [5, 6], decisions that even experts make based on vague intuition [66]. Even when users know what to analyze, the *scale of possible scenarios* presents another hurdle: as features and parameters increase, the space of possible scenarios–especially in **`SENSITIVITY`** and **`GOAL SEEK`** analyses–grows exponentially [22, 49], making exhaustive exploration infeasible. Strategies like greedy search [57] still force trade-offs between depth and breadth [47, 79], while new hypotheses and directions often emerge mid-analysis [51, 55], further expanding the search space. Furthermore, users struggle to *track evolving analyses*, as iteratively adjusting variables and exploring alternatives makes it difficult to follow which inputs led to which outputs, especially when comparing many scenarios via **`SCENARIO COMPARISON`**[10]. Reliance on memory or mental models [21, 85] increases cognitive overload and reduces the ability to extract actionable insights [2, 25].

Future Opportunities. Future systems could guide users in selecting and managing analyses through recommendation features and intelligent assistance [51]. Provenance tracking, as explored in workflow management systems such as VisTrails [29] and Kepler [59], could be adapted to what-if analysis to log user actions and hypotheses. Our PRAXA framework provides a starting point by serving structured and formal specifications of each analysis (inputs, operations, constraints, outputs) that can be used as metadata within such systems, enabling reproducibility and automated analysis pipelines that reuse prior workflows.

Challenges related to implementation of what-if analysis. Even if users know what scenarios to explore, actually performing and executing the what-if scenarios is often a *manual and tedious process*. It requires users to individually iterate on adjusting selected variables and re-running each configuration one at a time. Many systems rely on trial-and-error exploration with users manually manipulating input variables during **`SENSITIVITY`** analyses [34, 89] or re-running optimizations repeatedly in **`GOAL SEEK`** analyses to satisfy new constraints [63]. Even when scenarios are compared (**`SCENARIO COMPARISON`**) [5, 36, 98], outputs are static, leaving users to navigate and refine scenarios themselves. Setting up such configurations are often evolving, which may lead to users *struggling to observe consistency* between data and visualizations, since small changes can cascade unpredictably, disrupting other already explored scenarios [23, 111]. Further, many what-if systems are *rigidly implemented*, requiring tightly coupled models and variables, requiring substantial re-engineering for small modifications [85]. Finally, what-if analysis workflows are *fragmented across multiple tools* for modeling, visualization, and communication, increasing cognitive load [85] on users.

Future Opportunities. Formal intermediate representations of what-if analysis (e.g., JSON specifications, schemas, or SQL-like languages), as supported by our PRAXA framework, could modularize implementation and enable reusable plug-ins, making systems more flexible and easier to adapt. Such specifications would also help maintain consistency across data, models, and visualizations, reducing errors and keeping outputs synchronized. Building on this foundation, workflow management techniques could unify currently fragmented toolchains into cohesive systems that integrate modeling, scenario tracking, and visualization. Finally, mixed-initiative approaches that translate natural language interactions (captured via LLMs) into PRAXA specifications could automate pipelines and bridge low-level implementation with higher-level communication, reducing manual effort while supporting more adaptive and scalable workflows.

Trust and interpretability issues with what-if analysis. Another challenge highlighted in the literature in the adoption of what-if analysis is the lack of users' trust and confidence in the predictions and their ability to act on them. These issues arise from interconnected issues. First, *ground truth is often absent* to validate hypothetical scenarios [7, 73], which makes it difficult for users to assess the right model to use [22] or correct parameters to explore [85]. In high-stakes domains like healthcare, this leads to unreliable outputs, like for instance, expert users conducting **SENSITIVITY** analysis and novice users applying **GOAL SEEK** analysis may arrive at different conclusions due to confirmation bias or diagnostic overreach [93]. Second, explanation techniques used alongside what-if analysis, such as LIME, SHAP, and DiCE, are *fragile and sensitive to hyperparameters*, producing inconsistent outputs [51]. This is observed in **SENSITIVITY** analysis where any instability in the explanation layer can distort users' understanding of the model behavior. While not a flaw of what-if analysis itself, brittle interpretable methods can still undermine the interpretive process [96]. Finally, results often *lack actionability*. For instance, generated scenarios may ignore feasibility constraints [15], poorly communicate uncertainty among predictions [17], and results may not align with user goals or expertise [13, 98]. Therefore, in practice, users sometimes revert to simpler models for interpretability at the cost of analytical depth [49].

Future Opportunities. Addressing trust and interpretability requires progress on three fronts. First, methods for embedding causal inference into what-if analysis [9, 93, 105] can help distinguish correlation from causation, making results more replicable and reliable even in the absence of ground truth [95]. Second, explanation techniques need to become more robust either by stabilizing interpretability methods or developing new methods that provide consistent, reproducible explanations under varying parameter configurations. Third, systems should present results in ways that are both transparent and actionable, like for example by incorporating uncertainty-aware visualizations (e.g., confidence intervals in predictions and causal graphs) and by rendering decision trade-offs (via highlighting Pareto fronts, sensitivity summaries, and diffs of what changed across scenarios). Allowing users to progressively disclose and explore explanations instead of showing them scenarios or explanation information all at once, along with mixed-initiative guidance of natural language can help calibrate their confidence and more usable in decision-making. In all of these potential opportunities, PRAXA could play a central role if extended to explicitly capture an explanation layer (e.g., including causal assumptions, metadata about explanation tools and their stability, and interface-level details for progressive disclosure), thereby providing a structured foundation for building more trustworthy, transparent, and user-adaptive what-if analysis systems.

Challenges in visualizing and interacting with what-if analysis. Many what-if systems rely on static or limited visualizations that restrict user engagement, with much prior work (25%) focusing only on displaying predictive outputs [6, 25, 42, 93]. Even interactive systems typically restrict input manipulation to simple elements like sliders [5, 79], text fields [30, 101], or table edits [100, 105], offering little support for richer operations such as chaining analyses (e.g., feeding **GOAL SEEK** results into **SENSITIVITY**) or introducing custom constraints (e.g., limiting delivery time for frequent customers while keeping increases in shipping cost and labor within acceptable bounds).

Visualizing the outputs (see supplementary materials [1]) is equally challenging: high-dimensional data, multiple scenarios, and interdependent variables often lead to dense, overwhelming visuals. Existing approaches struggle to balance detail with usability, making it difficult to extract insights from the analyses. For example, SHAP-based sensitivity plots may obscure actionable findings [72], dimensionality reduction can distort variable relationships [88], and even simple bar charts for counterfactuals can introduce framing biases (i.e., overemphasizing certain outcomes) [44].

Future Opportunities. Scalable and interpretable visualizations are needed to support richer what-if analysis. Promising directions include advanced dimensionality reduction techniques [2, 14, 38] and surrogate models like LIME and SHAP with visual indicators for fast, interpretable approximations [51]. Such indicators could take the form of uncertainty overlaps (e.g., shaded overlays with blurred under-regions showing when different runs produce inconsistent outcomes) or consensus highlighting to emphasize features or neighborhoods that remain stable across perturbations. With mixed-initiative systems, such indicators can also be supported with textual overlays or uncertainty callouts [4, 36, 42] that would further enhance the analysis. Further, hierarchical or multi-scale views [84] that allow users to move fluidly between overview and details can be adopted for what-if analysis. Interaction design can also go beyond single sliders or tables toward multi-scenario workspaces, where dynamic linking and filtering are used to compare, and chain analyses side-by-side.

8 Conclusion

What-if analysis is a fundamental approach used across domains, yet its observed across works via different terminologies and applied in inconsistent ways, limiting its systematic progress. Through a literature review of 141 publications, we introduced PRAXA, a framework that clarifies the motivations, components and types of what-if analysis, and highlights the challenges in its implementation and application. By standardizing the vocabulary and unifying a structure, PRAXA provides a foundation for more consistent, transparent, and reusable analyses. It also opens opportunities for newer novel what-if analyses techniques, visualizations, and generalizable tools and LLM-driven interfaces that make what-if analysis more accessible, interpretable, and impactful across domains.

References

[1] Yongsu Ahn, Quinn K Wolter, Jonilyn Dick, Janet Dick, and Yu-Ru Lin. 2024. Interactive Counterfactual Exploration of Algorithmic Harms in Recommender Systems. In *2024 IEEE Visualization in Data Science (VDS)*. IEEE, 35–39.

[2] Daniel Atzberger, Tim Cech, Willy Scheibel, Jürgen Döllner, Michael Behrisch, and Tobias Schreck. 2024. A Large-Scale Sensitivity Analysis on Latent Embeddings and Dimensionality Reductions for Text Spatializations. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[3] Michael Beham, Wolfgang Herzner, M Eduard Gröller, and Johannes Kehrer. 2014. Cupid: Cluster-based exploration of geometry generators with parallel coordinates and radial trees. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1693–1702.

[4] Alexander Bendeck, Dennis Bromley, and Vidya Setlur. 2024. SlopeSeeker: a search tool for exploring a dataset of quantifiable trends. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. 817–836.

[5] Aditya Bhattacharya, Jeroen Ooge, Gregor Stiglic, and Katrien Verbert. 2023. Directive explanations for monitoring the risk of diabetes onset: introducing directive data-centric explanations and combinations to support what-if explorations. In *Proceedings of the 28th international conference on intelligent user interfaces*. 204–219.

[6] Yali Bian, Chris North, Eric Krokos, and Sarah Joseph. 2021. Semantic explanation of interactive dimensionality reduction. In *2021 IEEE Visualization Conference (VIS)*. IEEE, 26–30.

[7] Ayan Biswas, Guang Lin, Xiaotong Liu, and Han-Wei Shen. 2016. Visualization of time-varying weather ensembles across multiple resolutions. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 841–850.

[8] Maryam Booshehrian, Torsten Möller, Randall M Peterman, and Tamara Munzner. 2012. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1235–1244.

[9] David Borland, Arran Zeyu Wang, and David Gotz. 2024. Using counterfactuals to improve causal inferences from visualizations. *IEEE Computer Graphics and Applications* 44, 1 (2024), 95–104.

[10] Clara Bove, Marie-Jeanne Lesot, Charles Albert Tijus, and Marcin Detyniecki. 2023. Investigating the intelligibility of plural counterfactual examples for non-expert users: an explanation user interface proposition and user study. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 188–203.

[11] Kieran Browne, Ben Swift, and Terhi Nurmikko-Fuller. 2020. Camera adversaria. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–9.

[12] Chris Bryan, Xue Wu, Susan Mniszewski, and Kwan-Liu Ma. 2015. Integrating predictive analytics into a spatiotemporal epidemic simulation. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 17–24.

[13] Francisco Maria Calisto, João Fernandes, Margarida Morais, Carlos Santiago, João Maria Abrantes, Nuno Nunes, and Jacinto C Nascimento. 2023. Assertiveness-based agent communication for a personalized medicine on medical imaging diagnosis. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–20.

[14] Marco Cavallo and Çağatay Demiralp. 2018. A visual interaction framework for dimensionality reduction based data exploration. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[15] Furui Cheng, Yao Ming, and Huamin Qu. 2020. Dece: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1438–1447.

[16] SAP Community. [n. d.]. What-if analysis with SAP Analytics Cloud: Simulate your future business.

[17] Daniel Cornel, Artem Konev, Bernhard Sadransky, Zsolt Horvath, Eduard Gröller, and Jürgen Waser. 2015. Visualization of object-centered vulnerability to possible flood hazards. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 331–340.

[18] Anamaria Crisan, Nathan Butters, et al. 2024. Exploring Subjective Notions of Explainability through Counterfactual Visualization of Sentiment Analysis. In *2024 IEEE Evaluation and Beyond-Methodological Approaches for Visualization (BELIV)*. IEEE, 15–24.

[19] Anamaria Crisan and Michael Correll. 2021. User ex machina: Simulation as a design probe in human-in-the-loop text analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.

[20] Carolina Veiga Ferreira de Souza, Suzanna Maria Bonnet, Daniel de Oliveira, Marcio Cataldi, Fabio Miranda, and Marcos Lage. 2023. PROWIS: A Visual Approach for Building, Managing, and Analyzing Weather Simulation Ensembles at Runtime. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[21] Vaishali Dhanoa, Conny Walchshofer, Andreas Hinterreiter, Eduard Gröller, and Marc Streit. 2021. Fuzzy spreadsheet: Understanding and exploring uncertainties in tabular calculations. *IEEE Transactions on Visualization and Computer Graphics* 29, 2 (2021), 1463–1477.

[22] Dennis Dingen, Marcel van't Veer, Patrick Houthuizen, Eveline HJ Mestrom, Erik HHM Korsten, Arthur RA Bouwman, and Jarke Van Wijk. 2018. RegressionExplorer: Interactive exploration of logistic regression models with subgroup analysis. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 246–255.

[23] Mennatallah El-Assady, Rita Sevastjanova, Fabian Sperrle, Daniel Keim, and Christopher Collins. 2017. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 382–391.

[24] Mateus Espadoto, Gabriel Appleby, Ashley Suh, Dylan Cashman, Mingwei Li, Carlos Scheidegger, Erik W Anderson, Remco Chang, and Alexandru C Telea. 2021. UnProjection: Leveraging inverse-projections for visual analytics of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics* 29, 2 (2021), 1559–1572.

[25] Noyan Evirgen, Ruolin Wang, and Xiang'Anthony Chen. 2024. From text to pixels: Enhancing user understanding through text-to-image model explanations. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. 74–87.

[26] Microsoft Excel. [n. d.]. Introduction to What-If Analysis.

[27] Rebecca Faust, Carlos Scheidegger, and Chris North. 2023. Aardvark: Comparative visualization of data analysis scripts. In *2023 IEEE Visualization in Data Science (VDS)*. IEEE, 30–38.

[28] Nivan Ferreira, Marcos Lage, Harish Doraiswamy, Huy Vo, Luc Wilson, Heidi Werner, Muchan Park, and Cláudio Silva. 2015. Urbane: A 3D framework to support data driven decision making in urban development. In *2015 IEEE conference on visual analytics science and technology (VAST)*. IEEE, 97–104.

[29] Juliana Freire, Cláudio T Silva, Steven P Callahan, Emanuele Santos, Carlos E Scheidegger, and Huy T Vo. 2006. Managing rapidly-evolving scientific workflows. In *Provenance and Annotation of Data: International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*. Springer, 10–18.

[30] Sneha Gathani, Madelon Hulsebos, James Gale, Peter J Haas, and Çağatay Demiralp. 2021. Augmenting decision making via interactive what-if analysis. *arXiv preprint arXiv:2109.06160* (2021).

[31] Sneha Gathani, Zhicheng Liu, Peter J Haas, and Çağatay Demiralp. 2025. What-if Analysis for Business Professionals: Current Practices and Future Opportunities. (2025).

[32] Sneha Gathani, Zhicheng Liu, Peter J Haas, and Çağatay Demiralp. 2025. What-if Analysis for Business Users: Current Practices and Future Opportunities. *2025 CHI Conference on Human Factors in Computing Systems* (2025), arXiv–2212.

[33] Navid Ghaffarzadegan. 2021. Simulation-based what-if analysis for controlling the spread of Covid-19 in universities. *PloS one* 16, 2 (2021), e0246323.

[34] Reza Ghaiumy Anaraky, Kaileigh Angela Byrne, Pamela J Wisniewski, Xinru Page, and Bart Knijnenburg. 2021. To disclose or not to disclose: Examining the privacy decision-making processes of older vs. younger adults. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–14.

[35] Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. 2020. Vice: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th international conference on intelligent user interfaces*. 531–535.

[36] Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. 2021. Advice: Aggregated visual counterfactual explanations for machine learning model validation. In *2021 IEEE Visualization Conference (VIS)*. IEEE, 31–35.

[37] Jochen Görtler, Thilo Spinner, Dirk Streeb, Daniel Weiskopf, and Oliver Deussen. 2019. Uncertainty-aware principal component analysis. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 822–831.

- [38] David Hägele, Tim Krake, and Daniel Weiskopf. 2022. Uncertainty-aware multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 23–32.
- [39] Sagad Hamid, Adrian Derstroff, Sören Klemm, Quynh Quang Ngo, Xiaoyi Jiang, and Lars Linsen. 2019. Visual Ensemble Analysis to Study the Influence of Hyper-parameters on Training Deep Neural Networks.. In *MLVis@ EuroVis*. 19–23.
- [40] Jianing Hao, Qing Shi, Yilin Ye, and Wei Zeng. 2023. TimeTuner: diagnosing time representations for time-series forecasting with counterfactual explanations. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2023), 1183–1193.
- [41] Subhashis Hazarika, Haruki Hirasawa, Sookyung Kim, Kalai Ramea, Salva R Cachay, Peetak Mitra, Dipti Hingmire, Hansi Singh, and Phil J Rasch. 2023. HAiVA: Hybrid AI-assisted Visual Analysis Framework to Study the Effects of Cloud Properties on Climate Patterns. In *2023 IEEE Visualization and Visual Analytics (VIS)*. IEEE, 226–230.
- [42] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M Drucker. 2019. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.
- [43] Zhaosong Huang, Yafeng Lu, Elizabeth A. Mack, Wei Chen, and Ross Maciejewski. 2020. Exploring the Sensitivity of Choropleths under Attribute Uncertainty. *IEEE Transactions on Visualization and Computer Graphics* 26, 8 (2020), 2576–2590. doi:10.1109/TVCG.2019.2892483
- [44] Alex Kale, Yifan Wu, and Jessica Hullman. 2021. Causal support: Modeling causal inferences with visualizations. *IEEE transactions on visualization and computer graphics* 28, 1 (2021), 1150–1160.
- [45] Smiti Kaul, David Borland, Nan Cao, and David Gotz. 2021. Improving visualization interpretation using counterfactuals. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 998–1008.
- [46] Cheong Kim, Francis Joseph Costello, Kun Chang Lee, Yuan Li, and Chenyao Li. 2019. Predicting factors affecting adolescent obesity using general bayesian network and what-if analysis. *International journal of environmental research and public health* 16, 23 (2019), 4684.
- [47] Yuki Koyama, Daisuke Sakamoto, and Takeo Igarashi. 2014. Crowd-powered parameter analysis for visual design exploration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 65–74.
- [48] Alexander Kumpf, Josef Stumpfegger, Patrick Fabian Härtl, and Rüdiger Westermann. 2020. Visual Analysis of Multi-Parameter Distributions across Ensembles. *arXiv preprint arXiv:2007.15446* (2020).
- [49] Ripan Kumar Kundu, Osama Yahia Elsaid, Prasad Calyam, and Khaza Anuarul Hoque. 2023. Vr-lens: Super learning-based cybersickness detection and explainable ai-guided deployment in virtual reality. In *Proceedings of the 28th international conference on intelligent user interfaces*. 819–834.
- [50] Biagio La Rosa, Graziano Blasilli, Romain Bourqui, David Auber, Giuseppe Santucci, Roberto Capobianco, Enrico Bertini, Romain Giot, and Marco Angelini. 2023. State of the art of visual analytics for explainable deep learning. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, 319–355.
- [51] Sonia Laguna, Julian N Heidenreich, Jiugeng Sun, Nilüfer Cetin, Ibrahim Al-Hazwani, Udo Schlegel, Furui Cheng, and Mennatallah El-Assady. 2023. ExpLIMEable: A Visual Analytics Approach for Exploring LIME. In *2023 Workshop on Visual Analytics in Healthcare (VAHC)*. IEEE, 27–33.
- [52] Jungeun Lee, Sungnam Kim, Minki Cheon, Hyojin Ju, JaeEun Lee, and Inseok Hwang. 2022. SleepGuru: Personalized Sleep Planning System for Real-life Actionability and Negotiability. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–16.
- [53] François Lévesque, Louis Beaumier, and Thomas Hurtut. 2024. Pathways Explorer: Interactive Visualization of Climate Transition Scenarios. In *2024 IEEE Workshop on Energy Data Visualization (EnergyVis)*. IEEE, 29–33.
- [54] Guan Li, Junpeng Wang, Han-Wei Shen, Kaixin Chen, Guihua Shan, and Zhonghua Lu. 2020. Cnnpruner: Pruning convolutional neural networks with visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1364–1373.
- [55] Shusen Liu, Zhimin Li, Tao Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. 2018. Nlize: A perturbation-driven visual interrogation tool for analyzing and interpreting natural language inference models. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 651–660.
- [56] Wei Liu, Chris North, and Rebecca Faust. 2024. Visualizing Spatial Semantics of Dimensionally Reduced Text Embeddings. *arXiv preprint arXiv:2409.03949* (2024).
- [57] Yun-En Liu, Travis Mandel, Emma Brunskill, and Zoran Popović. 2014. Towards automatic experimentation of educational knowledge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3349–3358.
- [58] Martin Luboschik, Martin Röhlig, Arne T Bittig, Natalia Andrienko, Heidrun Schumann, and Christian Tominski. 2015. Feature-driven visual analytics of chaotic parameter-dependent movement. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 421–430.
- [59] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A Lee, Jing Tao, and Yang Zhao. 2006. Scientific workflow management and the Kepler system. *Concurrency and computation: Practice and experience* 18, 10 (2006), 1039–1065.
- [60] Wei Luo, Michael Steptoe, Zheng Chang, Robert Link, Leon Clarke, and Ross Maciejewski. 2017. Impact of spatial scales on the intercomparison of climate scenarios. *IEEE Computer Graphics and Applications* 37, 5 (2017), 40–49.
- [61] Yannick Metz, Dennis Ackermann, Daniel A Keim, and Maximilian T Fischer. 2024. Interactive Public Transport Infrastructure Analysis through Mobility Profiles: Making the Mobility Transition Transparent. In *2024 IEEE Visualization in Data Science (VDS)*. IEEE, 6–14.
- [62] Vladimir Molchanov and Lars Linsen. 2014. Interactive design of multidimensional data projection layout. (2014).
- [63] Hee-Seung Moon, Antti Oulasvirta, and Byungjoo Lee. 2023. Amortized inference with user simulations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.
- [64] Dinmukhammed Mukashev, Nimesha Ranasinghe, and Aditya Shekhar Nittala. 2023. Tacttongue: Prototyping electrotactile stimulations on the tongue. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[65] Daniel Orban, Daniel F Keefe, Ayan Biswas, James Ahrens, and David Rogers. 2018. Drag and track: A direct manipulation interface for contextualizing data instances within a continuous parameter space. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 256–266.

[66] Stephan Pajer, Marc Streit, Thomas Torsney-Weir, Florian Spechtenhauser, Torsten Möller, and Harald Piringer. 2016. Weightlifter: Visual weight space exploration for multi-criteria decision making. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 611–620.

[67] Cheonbok Park, Soyoung Yang, Inyoup Na, Sunghyo Chung, Sungbok Shin, Bum Chul Kwon, Deokgun Park, and Jaegul Choo. 2021. VATUN: Visual Analytics for Testing and Understanding Convolutional Neural Networks.. In *EuroVis (Short Papers)*. 7–11.

[68] Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 74, 18 pages. doi:10.1145/3526113.3545616

[69] Nikolaus Piccolotto, Markus Bögl, and Silvia Miksch. 2023. Visual parameter space exploration in time and space. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, e14785.

[70] Renata Georgia Raidou, Oscar Casares-Magaz, Ludvig P Muren, Uulke A Van der Heide, Jarle Rørvik, Marcel Breeuwer, and Anna Vilanova. 2016. Visual analysis of tumor control models for prediction of radiotherapy response. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 231–240.

[71] Jay Roberts and Theodoros Tsiligkaridis. 2021. Controllably Sparse Perturbations of Robust Classifiers for Explaining Predictions and Probing Learned Concepts. In *MLVis@ EuroVis*.

[72] Diego Rojo, Nyi Nyi Htun, Denis Parra, Robin De Croon, and Katrien Verbert. 2021. AHMoSe: A knowledge-based visual support system for selecting regression machine learning models. *Computers and Electronics in Agriculture* 187 (2021), 106183.

[73] Erik Rydow, Rita Borgo, Hui Fang, Thomas Torsney-Weir, Ben Swallow, Thibaud Porphyre, Cagatay Turkay, and Min Chen. 2022. Development and evaluation of two approaches of visual sensitivity analysis to support epidemiological modeling. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 1255–1265.

[74] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. 2019. What Do We Talk About When We Talk About Dashboards? *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 682–692. doi:10.1109/TVCG.2018.2864903

[75] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. 2014. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2161–2170.

[76] Daniel Seebacher, Tom Polk, Halldor Janetzko, Daniel A Keim, Tobias Schreck, and Manuel Stein. 2021. Investigating the Sketchplan: A novel way of identifying tactical behavior in massive soccer datasets. *IEEE Transactions on Visualization and Computer Graphics* 29, 4 (2021), 1920–1936.

[77] Omar Shaikh, Valentino Emil Chai, Michele Gelfand, Diyi Yang, and Michael S Bernstein. 2024. Rehearsal: Simulating conflict to teach conflict resolution. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–20.

[78] David A Shamma, Matthew L Lee, Alexandre LS Filipowicz, Laurent Denoue, Kate Glazko, Kalani Murakami, and Kent Lyons. 2022. EV Life: A Counterfactual Dashboard Towards Reducing Carbon Emissions of Automotive Behaviors. In *Companion Proceedings of the 27th International Conference on Intelligent User Interfaces*. 46–49.

[79] Jingyi Shen, Yuhan Duan, and Han-Wei Shen. 2024. Surroflow: A flow-based surrogate model for parameter space exploration and uncertainty quantification. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[80] Neng Shi, Jiayi Xu, Skylar W Wurster, Hanqi Guo, Jonathan Woodring, Luke P Van Roekel, and Han-Wei Shen. 2022. Gnn-surrogate: A hierarchical and adaptive graph neural network for parameter space exploration of unstructured-mesh ocean simulations. *IEEE Transactions on Visualization and Computer Graphics* 28, 6 (2022), 2301–2313.

[81] Rainer Splechtna, Krešimir Matković, Denis Gračanin, Mario Jelović, and Helwig Hauser. 2015. Interactive visual steering of hierarchical simulation ensembles. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 89–96.

[82] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush. 2018. S eq 2s eq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 353–363.

[83] Tableau. [n. d.]. Tableau Quick What-If.

[84] Wenbo Tao, Xiaoyu Liu, Yedi Wang, Leilani Battle, Çağatay Demiralp, Remco Chang, and Michael Stonebraker. 2019. Kyrix: Interactive pan/zoom visualizations at scale. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 529–540.

[85] Zain Tariq, Miro Mannino, Mai Le Xuan Anh, Whitney Bagge, Azza Abouzied, and Dennis Shasha. 2021. Planning epidemic interventions with EpiPolicy. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 894–909.

[86] Thomas Torsney-Weir, Ahmed Saad, Torsten Moller, Hans-Christian Hege, Britta Weber, Jean-Marc Verbavatz, and Steven Bergner. 2011. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 1892–1901.

[87] Ploii Tubsamon. [n. d.]. A step-by-step guide on creating What-if analysis in Power BI.

[88] Anjul Tyagi, Zhen Cao, Tyler Estro, Erez Zadok, and Klaus Mueller. 2019. Ice: An interactive configuration explorer for high dimensional categorical parameter spaces. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 23–34.

[89] Viny Saajan Victor, Andre Schmeißer, Heike Leitte, and Simone Gramsch. 2022. Visual parameter space analysis for optimizing the quality of industrial nonwovens. *IEEE Computer Graphics and Applications* 42, 2 (2022), 56–67.

[90] Tatiana Von Landesberger, Dieter W Fellner, and Roy A Ruddle. 2016. Visualization system requirements for data processing pipeline design and optimization. *IEEE Transactions on Visualization and Computer Graphics* 23, 8 (2016), 2028–2041.

[91] Arran Zeyu Wang, David Borland, and David Gotz. 2024. Beyond Correlation: Incorporating Counterfactual Guidance to Better Support Exploratory Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[92] Arran Zeyu Wang, David Borland, and David Gotz. 2024. An empirical study of counterfactual visualization to support visual causal inference. *Information Visualization* 23, 2 (2024), 197–214.

[93] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.

[94] Edward Jay Wang, Junyi Zhu, Mohit Jain, Tien-Jui Lee, Elliot Saba, Lama Nachman, and Shwetak N Patel. 2018. Seismo: Blood pressure monitoring using built-in smartphone accelerometer and camera. In *Proceedings of the 2018 CHI conference on human factors in computing Systems*. 1–9.

[95] Junpeng Wang, Wei Zhang, Hao Yang, Chin-Chia Michael Yeh, and Liang Wang. 2021. Visual analytics for rnn-based deep reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2021), 4141–4155.

[96] Xinru Wang and Ming Yin. 2021. Are explanations helpful? a comparative study of the effects of explanations in ai-assisted decision-making. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*. 318–328.

[97] Yunlong Wang, Priyadarshini Venkatesh, and Brian Y Lim. 2022. Interpretable directed diversity: Leveraging model explanations for iterative crowd ideation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–28.

[98] Zijie J Wang, Jennifer Wortman Vaughan, Rich Caruana, and Duen Horng Chau. 2023. Gam coach: Towards interactive and user-centered algorithmic recourse. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.

[99] Jürgen Waser, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Hrvoje Ribičić, Robert Carnecky, Patrick Kluding, and Benjamin Schindler. 2014. Many plans: Multidimensional ensembles for visual decision support in flood management. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 281–290.

[100] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 56–65.

[101] Sherry Wu, Hua Shen, Daniel S Weld, Jeffrey Heer, and Marco Tulio Ribeiro. 2023. Scattershot: Interactive in-context example curation for text transformation. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 353–367.

[102] Tongshuang Wu. 2021. Principles and Interactive Tools for Evaluating and Improving the Behavior of Natural Language Processing models. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–6.

[103] Yingcai Wu, Shixia Liu, Kai Yan, Mengchen Liu, and Fangzhao Wu. 2014. Opinionflow: Visual analysis of opinion diffusion on social media. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1763–1772.

[104] Tiankai Xie, Yuxin Ma, Hanghang Tong, My T Thai, and Ross Maciejewski. 2020. Auditing the sensitivity of graph-based ranking with visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1459–1469.

[105] Xiao Xie, Fan Du, and Yingcai Wu. 2020. A visual analytics approach for exploratory causal analysis: Exploration, validation, and applications. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1448–1458.

[106] Ke Xu, Meng Xia, Xing Mu, Yun Wang, and Nan Cao. 2018. Ensemblelens: Ensemble-based visual exploration of anomaly detection algorithms with multidimensional data. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 109–119.

[107] Shota Yamanaka and Hiroki Usuba. 2023. Tuning Endpoint-variability Parameters by Observed Error Rates to Obtain Better Prediction Accuracy of Pointing Misses. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[108] Jun Yuan, Brian Barr, Kyle Overton, and Enrico Bertini. 2022. Visual exploration of machine learning model behavior with hierarchical surrogate rule sets. *IEEE Transactions on Visualization and Computer Graphics* 30, 2 (2022), 1470–1488.

[109] Linping Yuan. 2023. Echoes in the gallery: A collaborative immersive analytics system for analyzing audience reactions in virtual reality exhibitions. (2023).

[110] Wei Zeng, Chengqiao Lin, Juncong Lin, Jincheng Jiang, Jiazhi Xia, Cagatay Turkay, and Wei Chen. 2020. Revisiting the modifiable areal unit problem in deep traffic prediction with visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 839–848.

[111] Emanuel Zgraggen, Robert Zeleznik, and Philipp Eichmann. 2016. Tableur: Handwritten spreadsheets. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2362–2368.

[112] Chenyang Zhang, Xiyuan Wang, Chuyi Zhao, Yijing Ren, Tianyu Zhang, Zhenhui Peng, Xiaomeng Fan, Xiaojuan Ma, and Quan Li. 2022. Promotionlens: Inspecting promotion strategies of online e-commerce via visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 767–777.

[113] Wencan Zhang and Brian Y Lim. 2022. Towards relatable explainable AI with the perceptual process. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–24.

[114] Liang Zhou and Charles Hansen. 2014. GuideME: Slice-guided Semiautomatic Multivariate Exploration of Volumes. In *Computer graphics forum*, Vol. 33. Wiley Online Library, 151–160.