Slicing, Chatting, and Refining: A Concept-Based Approach for Machine Learning Model Validation with *ConceptSlicer*

Xiaoyu Zhang*
xiaoyu.zhang@ai.ethz.ch
ETH AI Center
Zürich, Switzerland

Jorge Piazentin Ono
Robert Bosch Research and Technology Center
Sunnyvale, USA
jorge.piazentinono@us.bosch.com

Wenbin He
Robert Bosch Research and Technology Center
Sunnyvale, USA
Wenbin.He2@us.bosch.com

Liang Gou*
Splunk Technology
San Jose, CA, USA
lgou.psu@gmail.com

Mrinmaya Sachan
ETH Zürich
Zürich, Switzerland
mrinmaya.sachan@inf.ethz.ch

Kwan-Liu Ma
University of California, Davis
Davis, CA, USA
klma@ucdavis.edu

Liu Ren
Robert Bosch Research and Technology Center
Sunnyvale, USA
liu.ren@us.bosch.com

ABSTRACT

As machine learning (ML) gains wider adoption in real-world applications, the validation of ML models becomes fundamental for its productization, particularly in safety-critical applications. Recently, data slice finding has emerged as a popular method for validating ML models, but it requires additional metadata or cross-modal embeddings for the slices to be interpretable. We propose *ConceptSlicer*, an integrated workflow that facilitates the slicing of computer vision models using visual concepts. This approach breaks down the image dataset into interpretable visual concepts, serving as metadata in the slice finding process. Our system offers insights into model issues and enables a deeper understanding of computer vision models' strengths and weaknesses. We evaluate *ConceptSlicer* through interviews with eight domain experts and machine learning practitioners, and fine-tune the ML models based on their feedback. Our study also highlights varied attitudes towards large foundational models, encouraging contemplation of the challenges and opportunities presented by this technological advancement.

CCS CONCEPTS

• **Human-centered computing** → *Interactive systems and tools*; • **Computing methodologies** → **Machine learning**; **Model verification and validation**.

*This work was done when the authors worked with Robert Bosch Research and Technology Center.

IUI '24, March 18–21, 2024, Greenville, SC, USA

ACM ISBN 979-8-4007-0508-3/24/03
https://doi.org/10.1145/3640543.3645163

KEYWORDS

Data Slicing, Data-Centric AI, Human-in-the-loop

ACM Reference Format:
Xiaoyu Zhang, Jorge Piazentin Ono, Wenbin He, Liang Gou, Mrinmaya Sachan, Kwan-Liu Ma, and Liu Ren. 2024. Slicing, Chatting, and Refining: A Concept-Based Approach for Machine Learning Model Validation with *ConceptSlicer*. In *29th International Conference on Intelligent User Interfaces (IUI '24), March 18–21, 2024, Greenville, SC, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3640543.3645163

1 INTRODUCTION

In recent years, the advancement of machine learning techniques has significantly expanded the scope of problems that can be addressed through computational solutions. Notably, machine learning has been applied in various critical tasks, including but not limited to intelligent transportation [8, 10, 22], medical image processing [5, 23, 38], and e-commerce [54, 60]. Given the stringent demands for effectiveness and reliability in these scenarios, it becomes imperative to ensure the validity of machine learning models, particularly in terms of their robustness in critical edge cases [63].

Among existing fine-grained evaluation approaches, data slice finding [3, 14, 45, 48, 51] stands as an efficient method for validating machine learning models by uncovering potential issues on data subsets. However, achieving transparency and interpretability in slice finding often necessitates the incorporation of additional metadata or cross-modal embeddings to interpret the outcomes and align them with domain experts and ML practitioners' knowledge. Despite these efforts, domain practitioners still require additional support to comprehend why the model fails on these slices before deciding which slice to prioritize for model optimization. When it comes to the model optimization stage, data augmentation is a widely adopted strategy to rectify model biases and performance issues. Nevertheless, gathering the appropriate data to mitigate a model issue remains a resource-intensive process, both in terms of

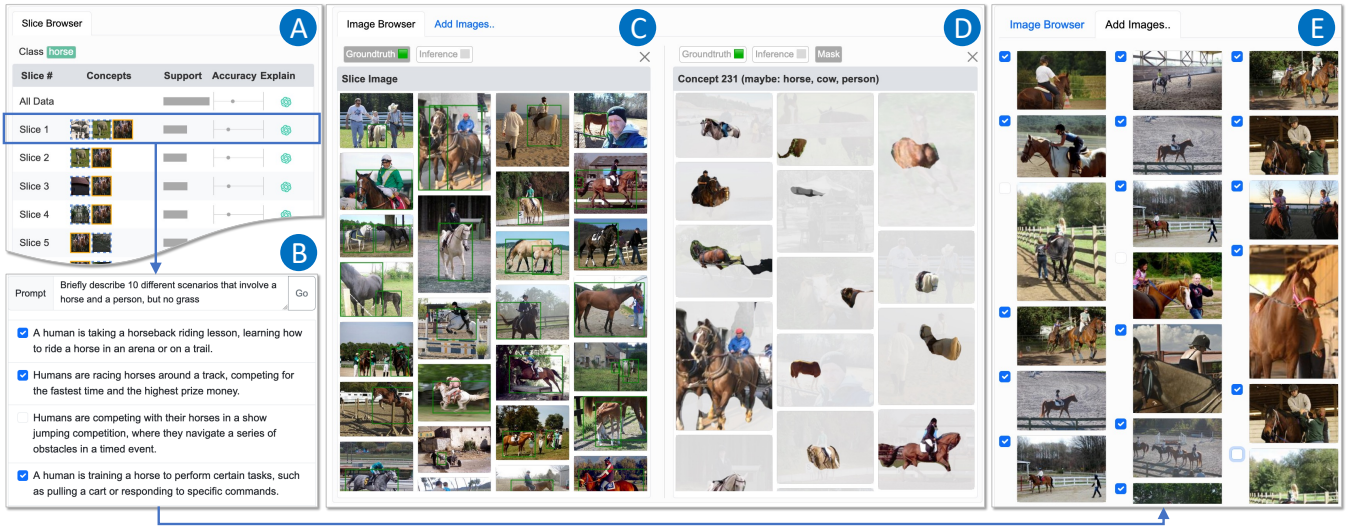


Figure 1: *ConceptSlicer*can be used to alleviate model defects caused by complicated cases involving spurious correlations and object overlapping. The (A) Slice Browser shows that an object detection model that detects objects of class "horse" has the worst performance on Slice 1. An examination of image and concept details in both the (C) Image Browser and (D) Concept Browser suggests that this issue may be attributed to a spurious correlation between horses and grass, as well as the overlap between horse and human torsos. To rectify this problem, we engage ChatGPT to generate "scenarios with horses and a person, but without grass" and carefully select relevant scenarios (B). Eventually, we eliminate irrelevant images from the retrieved images (E), and then export the supplementary images to augment our dataset and facilitate model fine-tuning.

cost and time. This highlights the need for more efficient methodologies in data collection and model optimization.

To address these challenges, we have developed an integrated workflow, *ConceptSlicer*, and have created a prototype system based on this workflow. *ConceptSlicer* is designed to assist machine learning researchers and engineers involved in computer vision tasks, specifically focusing on diagnosing object detection models and developing more effective data augmentation strategies. Unlike many existing slice finding solutions [11, 32, 39, 63], our system does not require additional metadata or cross-modal embeddings as input. Instead, it leverages the semantic information inherent in the images themselves and generates visual concepts using a self-supervised segmentation model. Using these visual concepts as metadata, *ConceptSlicer* can perform and present the slice finding results to users through a variety of visualizations and interactions. Additionally, we introduce a novel approach to enable users to retrieve images from a supplementary dataset for data augmentation. This functionality allows users to interact with ChatGPT [42] to generate textual descriptions of the desired supplementary image scenarios. Leveraging cross-modal embeddings from CLIP [46], *ConceptSlicer* then retrieves images from the supplementary dataset based on these text descriptions.

We present a use case to demonstrate how *ConceptSlicer* offers insights into model issues. Furthermore, we assess *ConceptSlicer*by seeking feedback and model optimization solutions from eight domain experts and ML practitioners. The results affirm that *ConceptSlicer*equips researchers and practitioners with a more profound understanding of the strengths and weaknesses of the computer vision models. This insight fosters better decision-making and problem-solving. Additionally, we observe a range of attitudes toward the emergence of large foundational models. This diversity in perspectives prompts contemplation of the opportunities and challenges introduced by this new wave of technology.

In summary, our primary contributions comprise the following:

- **An integrated workflow for ML model validation:** We introduce an innovative integrated workflow *ConceptSlicer*, which leverages the knowledge from large foundational models to validate and optimize computer vision models. This approach eliminates the need for additional metadata or cross-modal embeddings during the data slice finding process. This workflow is implemented as a web-based system that facilitates data slice browsing, failure case diagnosis, and model optimization.
- **A methodology for metadata-free data slice finding**: We introduce a pioneering methodology that combines visual concept extraction and scalable data slice finding. Our visual concept extraction method provides a detailed interpretation of images based on visual concepts, effectively eliminating the need for additional metadata. Then, we apply a state-of-the-art data slice finding method that identifies and extracts problematic data slices based on these visual concepts. This innovative approach fosters a more comprehensive understanding of the data slice and facilitates the identification and resolution of potential issues within the slices.
- **A methodology for user-driven data augmentation**: We propose a novel methodology that augments the training dataset with insights gathered from problematic data slices. This process involves the identification of problematic slices, which are then

incorporated into the training dataset. This augmented dataset is utilized to train a new model, which can address the problems identified in the most critical data slices.
- **Use cases and expert review study:** We present a use case and an expert review study, both of which exemplify the practical application of *ConceptSlicer* and gather valuable feedback from domain practitioners. Additionally, our paper delves into deeper insights and reflections concerning domain experts and ML practitioners' attitudes and interaction patterns in relation to large foundational models.

2 RELATED WORK

2.1 Slice-based Model Validation and Optimization

Data slice finding, the process of identifying specific data subgroups where machine learning models exhibit subpar performance, as described in Barash et al.'s work [3], serves as a common approach for fine-grained model evaluation. Typically, this process entails a combination of automated detection and the extraction of domain expertise from human knowledge, although the emphasis on each aspect may vary. Existing tools like SliceLine [48], SliceFinder [14], DivExplorer [45], and GEORGE [51] offer automated slice detection with minimal to no human intervention. In contrast, human-in-the-loop solutions such as SliceTeller [63], Visual Auditor [39], CoFact [32], Mandoline [11], VLSlice[50] and Deblinder [7] have emerged in recent years, which strive to leverage domain expertise by creating interactive visual analytics systems and gathering expert input for slice prioritization. The shortcoming is that the majority of them necessitate either additional metadata or human participation to complete the slice finding process. In this work, we offer a comprehensive slice finding solution that eliminates the need for additional metadata or human intervention. We integrate SliceLine [48] for its minimal parameter tuning requirements and high search efficiency in larger scale according to the comprehensive comparison of prevalent slice finding algorithms in [49]. Meanwhile, *ConceptSlicer* provides the capability for humans to inspect the results and make optimization decisions afterward.

In addition to model validation, our workflow also incorporates steps dedicated to enhancing model robustness on a per-slice basis, thus completing the model optimization loop. Existing solutions in the field of robustness research primarily target mitigating the adverse effects stemming from spurious correlations [1, 35, 40], which often lead to performance degradation under distributional shifts [18, 36]. Notably, researchers have proposed various approaches, including Group Distributionally Robust Optimization (Group DRO)[27, 44], sample reweighting[11, 12, 20, 61], and data augmentation methods, to boost model robustness. Some recent studies, such as PromptAid [37] and Visual Auditor [39], emphasize practical applications by tackling real-world challenges like model bias detection and language model evaluation. In our work, we employ a data augmentation combined with fine-tuning method to streamline the model optimization process, reducing time costs and better aligning with real-world requirements.

2.2 Visual Concept for eXplainable AI

In recent years, the concept-level interpretation of machine learning model behaviors has been widely used in eXplainable AI (XAI) to interpret and explain what has been learned by a CNN model [1, 13, 21, 28, 33]. According to Bau et al. [4], concepts are human interpretable abstractions extracted from images, which are typically represented as image segments with semantic meanings. Most of the existing work directly uses visual concepts and requires human intervention to build a connection between these concepts and model behaviors [65], spurious correlations [1, 34], or conceptual overlapping between classes [56]. In contrast, *ConceptSlicer* uses the concepts as input for a slice-finding algorithm and eliminates the need for human effort in exploring concept and model behavior at the current stage, saving their time for more critical tasks.

Depending on the scale of the data and the specific objectives, visual concepts can be generated through either manual or data-driven approaches. For example, previous works [4, 33] have employed human-specified concepts to understand user interest or assess the interpretability of neural networks. More recently, a variety of data-driven approaches have emerged for concept extraction from images, employing superpixels [21], prototypes [9], and dictionaries of object parts [29]. In our research, we leverage the concept generation method introduced in [25], which extends the embeddings generated by pre-trained vision-language models (e.g., CLIP [47]) from the image level down to the pixel level. This approach enables us to learn fine-grained and sharper pixel embeddings through self-supervision, outperforming methods like MaskCLIP [66] and MaskCLIP+ [66] in this regard.

2.3 Large Foundational Models

Since OpenAI launched ChatGPT [42] at the end of 2022, there has been a surge of exploration and expansion in the realm of large foundational models, particularly within the domain of large language models, spanning both algorithmic advancements and practical applications. The initial iteration of ChatGPT was based on the Generative Pre-trained Transformer 3 (GPT-3) model [6], which was fine-tuned specifically for generating conversational responses. It was subsequently updated to GPT-3.5 [41] to further enhance its capabilities. In parallel, other LLMs like Google PaLM 2 [2], and Meta LLaMA 2 [53] have emerged, each showcasing diverse capabilities tailored to different data types and task complexities. Concurrently, the emergence of more lightweight and open-sourced language models, such as Stanford's Alpaca [52] and UC Berkeley's Koala [19], has illustrated that smaller-architecture models can also achieve competitive performance if trained on the right data.

In addition to the advancements in large language models, there have been significant developments in vision-language models. Models like CLIP [47] and GPT-4 [43], which are trained on extensive image-text datasets, have opened up exciting possibilities in various domains. These vision-language models have demonstrated their potential across a spectrum of downstream tasks, including image synthesis [31, 55] to out-of-distribution detection [15] and object detection [24]. For *ConceptSlicer*, we employ the ChatGPT API to facilitate the chat interface and leverage CLIP [47] for image retrieval using text input. This integration offers a more seamless

and user-friendly experience. In Sec. 6, we will delve deeper into our insights and reflections based on our experience with this tool.

3 *CONCEPTSLICER*

3.1 Design Requirements

Our system requirements have been derived from our interviews with machine learning model developers, along with insights gathered from the work of Zhang et al. [63]. We detail these requirements below:

- [**R1**] **Slice Identification and Overview**: Our system should automatically detect data slices where the vision model's performance dips below the average, subsequently offering a comprehensive overview of the model's metrics. It is crucial that the data samples within each slice share characteristics that are easily interpretable by humans. Furthermore, given the limited availability of annotated data, the identification of these slices should depend exclusively on the information inherent in the data itself, without the necessity for additional metadata.
- [**R2**] **Root Cause Analysis**: The system should help users to profile data slices and diagnose the reasons behind model failure on these slices. More specifically, users should be able to delve into the data samples within these slices and analyze the inference results produced by the model. Furthermore, users should grasp the context of each slice, which would aid them in prioritizing the slices and offering guidance during the optimization phase.
- [**R3**] **Slice-Based Model Optimization**: The users' primary goal is to train a ML model that exhibits satisfactory performance on crucial data slices. To fulfill this goal, our system needs to provide an all-encompassing solution and the necessary support to complete the machine learning optimization loop. Within the scope of this paper, there are two specific sub-requirements related to the definition, search, and refinement of training data based on visual concepts.
  R3-1 Task Prioritization and Customization: In practical applications, users frequently need to balance the performance across various scenarios to concentrate on critical use cases [63]. Consequently, they should be able to guide the model optimization process based on their domain knowledge and insights into the crucial slices. The system should assist users in effectively converting their understanding of failure cases into actionable insights for further model improvement.
  R3-2 Customized Data Augmentation: For users who choose to optimize the model via data augmentation, it's crucial to efficiently expand the training data by selecting new samples that fulfills specific data requirements. This data may not be part of the current dataset, but it may possess characteristics similar to the critical data slices and scenarios selected by the user. The system should be able to augment the data based on user specifications and provide interactive methods for the user to explore this enhanced dataset.

3.2 Workflow

To meet the design requirements described in Sec. 3.1, we developed a workflow that optimizes the performance of object detection models with data augmentation. The workflow, illustrated in Fig.2, consists of three stages that operate in an iterative manner. We name each task based on its key operation and organize the technical details as follows:

(1) **Slicing**: This stage involves finding under-performing data slices based on the model performance data and metadata generated using self-supervision techniques (Sec. 3.3).
(2) **Chatting**: This stage involves generating natural language descriptions of the detected data slices by chatting with ChatGPT to support failure mode analysis. (Sec. 3.4).
(3) **Refining**: This stage involves augmenting the training dataset by querying similar data from the supplementary dataset and retraining the model (Sec. 3.5).

3.3 *Slicing*: Concept-Based Slice Finding

A primary obstacle in determining interpretable slices is the requirement for interpretable metadata, which typically demands high-quality and labor-intensive manual annotation [63]. In the field of computer vision, we address this challenge by employing interpretable visual concepts that are automatically identified using self-supervised learning approaches [26, 62]. To reflect the image content, we collect all concepts present within the dataset and use binary encoding to indicate their presence in each image – assigning 1 if a concept is present, and 0 otherwise. This approach allows us to generate metadata for the dataset without any external input. As illustrated in Fig. 1, performing slice finding on such metadata enables the effective grouping of underperforming images that share similar visual elements.

However, such concept-based metadata is typically high-dimensional and extremely sparse, leading to excessive computational challenges for the slice finding process. For instance, we identified 512 concepts from the Pascal dataset [16] and discovered that depth-first slice finding solutions, such as DivExplorer [45], became impractical due to the substantial computational load. Therefore, we adopted SliceLine [48], a breadth-first slice finding toolkit where the search time is proportional to search depth (i.e., the maximum number of concepts defining the slice). It is noteworthy that our workflow is flexible to different slice finding methods. Alternate approaches, such as SliceFinder [14], can be easily integrated into the system to replace SliceLine. Our experiments revealed that SliceLine could efficiently identify slices characterized by at most three levels (i.e. three concepts) within a reasonable time frame. To further expedite the process, we narrowed our focus to a single class and eliminated any unrelated images and concepts during a preprocessing phase. Specifically, we discarded the images that did not contain any instance of the current class under examination, and removed the columns corresponding to the concepts that were not present in the remaining images from the metadata.

The breadth-first searching approach presents an additional challenge due to its limited support for efficient pruning compared to depth-first searching. Consequently, the number of identified slices tends to be large and highly similar. To address this issue, we computed a Jaccard similarity matrix for the identified slices and filtered

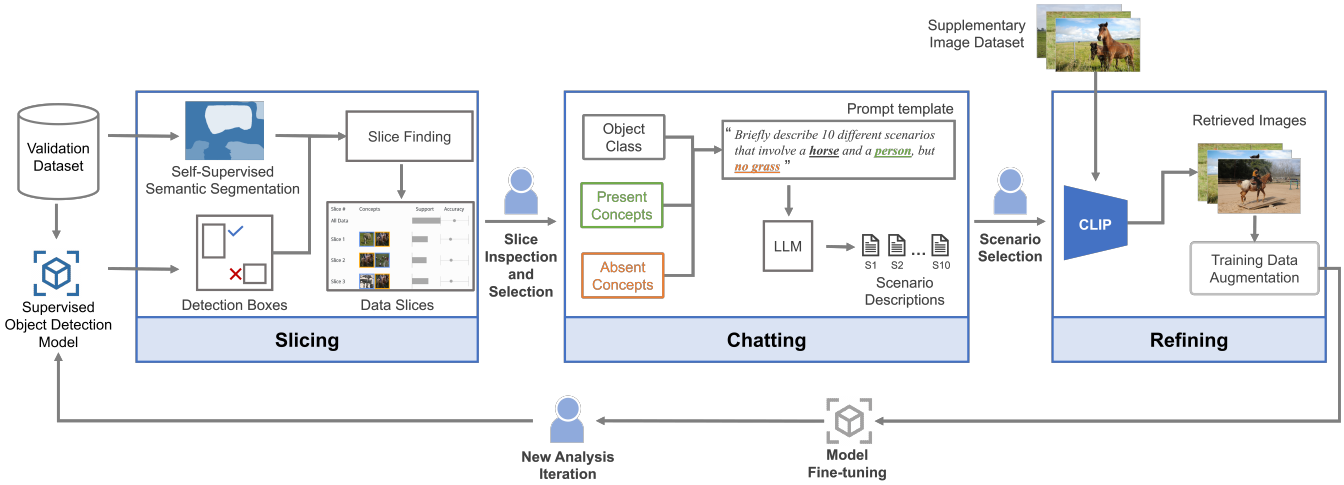


Figure 2: Workflow for the *ConceptSlicer* system. The process starts with the "Slicing" stage, where images undergo decomposition into semantic segments. These segments serve as a basis for pinpointing problematic data slices utilizing data slice finding techniques. Following this, the "Chatting" stage enables users to engage in discussions with a Large Language Model (LLM) to gain a deeper comprehension of the edge case scenarios represented by the data slices and uncover possible instances where these scenarios could occur. The final stage, "Refining", involves users retrieving new image data embedded with information beneficial in addressing the edge case scenarios identified in the problematic data slices.

out those with a similarity exceeding an empirically determined threshold. Moreover, we performed additional filtering based on slice size and *average accuracy* to select the most critical slices for further analysis.

Table 1: Examples of chatGPT-generated scenario descriptions from prompt "*Briefly describe 10 different scenarios that involve a horse and a person, but no grass*" and their similarity to slice 1 in Fig. 1.

Scenario Descriptions	Similarity
1. A human is taking a horseback riding lesson, learning how to ride a horse in an arena or on a trail.	0.2570
2. Humans are racing horses around a track, competing for the fastest time and the highest prize money.	0.2481
3. Humans are competing with their horses in a show jumping competition, where they navigate a series of obstacles in a timed event.	0.2415
4. A human is training a horse to perform certain tasks, such as pulling a cart or responding to specific commands.	0.2407
5. Humans are working with horses in a therapeutic setting, using the horses' calming presence to help individuals with various mental health conditions.	0.2385
6. Humans are involved in the process of breeding horses, selecting specific horses to produce offspring with desirable traits.	0.2336
7. Humans are using horses to transport equipment and supplies on a camping or hunting trip.	0.2307
8. Humans are rescuing horses from neglectful or abusive situations and rehabilitating them for adoption or sanctuary.	0.2292
9. Humans are showing off their horses in a competition or exhibition, demonstrating their beauty, agility, and training.	0.2037
10. Humans are taking a leisurely carriage ride, drawn by a horse, as a romantic or nostalgic activity.	0.2008

3.4 *Chatting*: Slice Explanation with ChatGPT

While interpretable visual concepts can highlight a few key elements of each slice, users must still examine samples within the slice to grasp its primary theme and discern the reason for the model's failure. Drawing inspiration from recent advancements in ChatGPT, we facilitate this process by employing it to generate scenario descriptions for a given slice. The input prompt for ChatGPT [42] is constructed by merging the class name and concept labels with a predefined template. For instance, the prompt for slice 1 in Fig.1 is "*Briefly describe 10 different scenarios that involve a horse and a person, but no grass*". An example output of ChatGPT, powered by GPT-4 [43], is presented in Table1.

Users can select from the generated scenario descriptions using interactive widgets, as described in Sec.3.6.1, before employing them to retrieve supplementary image datasets (Sec.3.5). To streamline this process, we arrange the scenario descriptions according to their semantic similarity to the images in the corresponding slice. This is accomplished by obtaining unified embeddings for both the scenario descriptions and the images within the slices using OpenCLIP [30], an open-source implementation of CLIP [46]. We then sort the descriptions based on the average cosine similarity between each pair (Algorithm 1). For example, Table 1 is sorted in descending order (from most to least similar) using this method.

3.5 *Refining*: Model Optimization with Data Augmentation

To complete the machine learning model optimization loop, *ConceptSlicer* enables users to retrieve data from supplementary datasets and augment the training set. Similarly, we utilize OpenCLIP to convert the images in the supplementary dataset into word embeddings and perform the query process by comparing the cosine

Algorithm 1 Computing Scenario Description Similarity

Input: *SliceImages*, *Descriptions*
Output: *DescriptionSimilarity*
for each *description* ∈ *Descriptions* **do**
  similarities ← [ ]
  $V_{description} \leftarrow TextEncoder\,(description)$
  for each *image* ∈ *SliceImages* **do**
    $V_{image} \leftarrow ImageEncoder\,(image)$
    $similarity \leftarrow cos\_sim\left(V_{description}, V_{image}\right)$
    similarities.append (*similarity*)
  end for
  description_similarity ← *avg* (*similarities*)
end for

similarity between these embeddings and the word embeddings of the scenario descriptions chosen by users. The most similar images are displayed in the system interface, allowing users to select and export them for retraining purposes. We illustrate this process in Fig. 3.

Finally, we utilize the user-selected images to augment the training data and subsequently fine-tune the machine learning model using the refined dataset. Specifically, for each iteration, we curate a new dataset by merging the user-selected images with the original training data. Then we fine-tune the original model on the new dataset for one epoch. This iterative process helps improve the model's performance and address its weaknesses.

3.6 System Design

3.6.1 Slice Browsing and Failure Diagnosing. We've designed a cohesive interface comprising three panels — a Slice Browser Panel (Fig. 1-A), an Image Browser Panel (Fig. 1-C), and a Concept Browser (Fig. 1-D) — to assist users in comprehending the under-performing data slices identified in Section 3.3 and analyzing why the model encounters difficulties with them. Our workflow begins with the Slice Browser Panel, featuring a table showcasing the poorest-performing data slices. Each row in the table furnishes essential details about a slice, including its index, representative concepts, support, accuracy, and a button to trigger the ChatGPT explanation for that specific slice. Here we define $acc = \min_{bbox1,bbox2,\ldots,bbox_T}(IOU)$ to represent the model's poorest performance in cases involving multiple target objects in the image. For a more intuitive grasp of the slice's images, users can simply click on a row, prompting the display of sample images from that slice in the Image Browser Panel. Additionally, we provide users the flexibility to toggle the visibility of ground truth and model inference bounding boxes, enhancing their understanding of prediction deviations.

3.6.2 Slice Portrayal. To offer an efficient, high-level understanding of the slice and provide hints about possible reasons for the model's failures, we summarize each slice using a selection of representative concepts in addition to supporting sample images browsing. Each concept is depicted as a representative thumbnail, with solid orange borders indicating presence and dotted blue borders indicating absence. For more in-depth insight, *ConceptSlicer* displays a tooltip upon hovering over a concept, revealing the concept

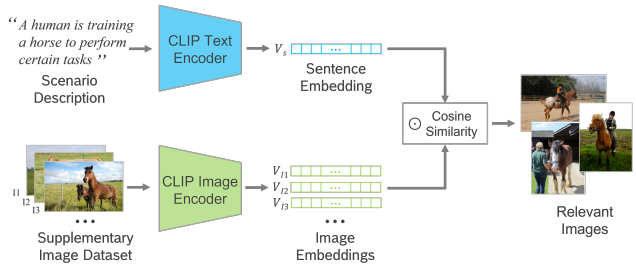


Figure 3: Image Retrieval Pipeline: In the first step, we apply the Vision+Language foundation model, CLIP, to retrieve images in accordance with the Scenario Descriptions generated by *ConceptSlicer*. CLIP is used to derive text embeddings from these descriptions and to generate image embeddings for a new image dataset. Subsequently, we employ the CLIP image retrieval approach, aligning the scenario descriptions with the most suitable images using a cosine similarity metric.

index, reference keywords, and an enlarged thumbnail. Recognizing the crucial role of accurate concept perception in understanding model failures, we go a step further by presenting sample images for each concept in the Concept Browser Panel (Fig. 1-D). Aside from viewing ground truth and model inference bounding boxes, *ConceptSlicer* offers users the option to show or hide a mask for each sample image, highlighting the specific image area corresponding to the concept. For instance, in Fig. 1, we can observe that Concept 124 primarily relates to the ground or grass in the image, potentially serving as a spurious feature for the model's "horse" detection. It's important to note that the sample images for a concept may not necessarily originate from the current slice, as the representative concept might be an absent concept in the current slice.

In our workflow, it is also essential to generate natural language descriptions for the currently inspected slice based on user understanding to facilitate the subsequent retrieval of images from the supplementary dataset. To facilitate this process, we have integrated a chat window within the slice browser panel (Fig. 1-B). The chat window begins with a prompt textbox, activated by the user beneath the slice they are examining. By default, this textbox displays text generated based on the representative concepts of the slice (see details in Section 3.4). Meanwhile, we offer users complete flexibility to edit the text or even entirely rewrite the sentence to align it with their understanding of the slice they are exploring. Once users are content with the prompt and send the request, a list of ten representative scenarios that might be present in the current slice will be presented, with the first three pre-selected by default. Users can then choose scenarios based on their understanding of the slice and their proposed solutions for augmenting the dataset. These scenario descriptions serve a dual purpose: they assist users in comprehending the current slice's content and provide guidance for directing the refining phase of the workflow. If users find the scenario descriptions unsatisfactory, they can iterate over the prompt editing and scenario selection steps as needed.

3.6.3 Optimization Support. To advance toward the ultimate objective of optimizing the model, *ConceptSlicer* assists users in gathering

Figure 4: Illustration of the failure case with the "car" class. In the case of the 'car' class, underperforming slices in the (a) Slice Browser are primarily associated with the absence of concept 440. Our vision-language model suggests that this concept may represent objects like *car*, *truck*, and *tvmonitor* (b). Upon closer examination of concept images, we find that this concept is closely related to a critical car component: windows (c). Consequently, the model performs poorly in images where windows are not visible due to the viewing angle or when cars are too small to discern the windows (d).

a list of supplementary images they believe will enhance model retraining. Upon the user's satisfaction with their selected scenario description, the system will retrieve supplementary images, adhering to the methodology outlined in Sec. 3.5. These images are presented in a grid view, allowing users to browse and select the ones they wish to incorporate into the fine-tuning process (Fig. 1-E). As described in Section 3.5, the chatting and refining stages can occur iteratively until users are satisfied with the set of supplementary images. At that point, they can choose to export the list of images for model optimization. The retraining process of the model occurs offline, providing the user with the flexibility to choose their preferred tool, such as PyTorch or TensorFlow. Users can import the new training results back into *ConceptSlicer* and initiate a new round of analysis once the training process is completed.

4 USE CASES

4.1 Use Scenario 1: Addressing Challenging Cases from a Machine Learning Model

The first scenario connects the examples used in Sec. 3 and demonstrates how *ConceptSlicer* can help alleviate model defects caused by complicated cases involving spurious correlations and object overlapping. This scenario involve the "horse" class. After importing the model performance data into the system, the user notice that the model has the worst performance on *Slice 1*, defined by two absent concepts, concept 102 (maybe *person, torso, arm*) and concept 124 (maybe *sheep, cow, grass*), and a present concept 231 (maybe *horse, cow, person*). After checking the suggested keywords and browsing the sample images of these concepts, they found that most images in the slice depicts people riding horses on the ground without grass (Fig. 1-C). Thus they infer that there might be a spurious correlation between horse and grass. Moreover, they observe that in most of such scenarios, people overlap with horses and the bounding boxes of the inference results unavoidably include both horse and human torsos. The user inferred that this might cause the confusion for the model, which eventually leads to wrong prediction or low confidence. Based on such inference, they instructed *ConceptSlicer* to request for scenario descriptions from ChatGPT with the prompt "*Briefly describe 10 different scenarios that involve a horse and a person, but no grass*" and received the results shown in Table 1 (E5). They selected the initial three sentences for their encapsulation of the most interesting scenarios that align with the prompt. Subsequently, they directed CLIP to fetch the images corresponding to these selected narratives. The initial results are shown in Fig. 1-E. After browsing the retrieved images and removing the irrelevant ones (those without horses or with low quality), the user exported the supplementary images and fine-tuned the model with the augmented dataset. The processes presented in this use case draw inspiration from the techniques employed by experts during the user study. Our results demonstrate that the model's performance saw significant improvement with the addition of only a handful of samples. This validates the efficacy

(a) Scenario: "*A car covered in a thick layer of fresh snow after a heavy winter storm*."

(b) Scenario: "*A car chase scene at night, where the windows are heavily tinted, adding to the suspense and mystery surrounding the pursuit."*

(c) Scenario: "*In a busy city street, a small car is parked in a parking garage, hidden from view as pedestrians walk by, unaware of its presence*."

Figure 5: Supplementary images to augment the dataset for the "car" class. These images "*involve cars but car windows are not visible*", and thus address the critical concept absence problem described in Sec. 4.2.

of our approach, showcasing its potential for enhancing outcomes with minimal sample augmentation. More feedback for the experts and ML practitioners and evaluation results are available in Sec. 5.

4.2 Use Scenario 2: Mitigating the Impact of Absent Critical Concepts

The second scenario illustrates the use of our system to augment the training set with images containing edge cases, thereby enhancing the model's robustness. When using *ConceptSlicer* to profile the "car" class, the user noticed that the model's overall performance was not acceptable ($acc = 0.32$). In their investigation of the slices in the Slice Browser, they discovered that the underperforming slices were consistently associated with the absence of the same concept, 440 (as shown in Fig. 4 (a)). This prompted them to delve deeper into why the absence of this particular concept had a recurring impact on model performance. Upon referring to the Concept View and highlighting concept 440, they realized that this concept bore a significant relevance to a critical component of the car: the windows (as shown in Fig. 4 (c)). In the majority of the slices where concept 440 was absent, the windows were either not visible due to the viewing angle or because the cars were too small to discern the windows (as depicted in Fig. 4 (d)). To mitigate the decline in model performance caused by the absence of this concept or due to poor image quality, the user decided to incorporate additional images that could enhance the model's robustness and accuracy.

They prompted ChatGPT to describe scenarios that "*involve cars but car windows are not visible*" and selected the following three scenarios for image retrieval:

- *A car covered in a thick layer of fresh snow after a heavy winter storm.*
- *A car chase scene at night, where the windows are heavily tinted, adding to the suspense and mystery surrounding the pursuit.*
- *In a busy city street, a small car is parked in an underground parking garage, hidden from view as pedestrians walk by, unaware of its presence.*

As shown in Fig. 5, *ConceptSlicer* retrieved relevant images to these scenarios, which increased the coverage of the dataset and improved the model's overall performance.

5 EXPERT REVIEW

5.1 Participants

Our prototype was reviewed by 8 domain experts or ML practitioners (4 female, 4 male) aged 25-44 years, consisting of 5 senior Ph.D. students and 3 research scientists with Ph.D. degrees. All participants possessed sufficient experience in utilizing machine learning models for image processing tasks and were well-versed in machine learning, making them suitable for providing expertise feedback on *ConceptSlicer*. Specifically, one participant reported over 10 years of involvement in machine-learning-related research, six had experience spanning 3-10 years, and one for 1-3 years. Among the 8 participants, six reported a familiarity level of 5 or more (on a 7-point Likert Scale) with foundational models.

5.2 Setup and Procedure

Given the diverse availability and geographical spread of participants, the study was carried out remotely involving all research scientists, while it was conducted in person exclusively with Ph.D. students. However, we maintained consistency by ensuring that all participants accessed *ConceptSlicer* using identical devices (14 inch laptop with Intel(R) Core(TM) i5-10310U CPU and 16GB RAM) and

the Firefox browser running on the Windows operating system. To capture the thought process, we requested participants to adhere to the "think aloud" protocol while we recorded both audio and video of their interactions throughout the task. As a token of appreciation for their involvement, each participant was provided with a 10 Amazon gift card.

The setup, tasks, and duration of the semi-structured, open-ended expert interview were established following a pilot study conducted with a co-author. Prior to the online session, participants were instructed to review the documentation for *ConceptSlicer* and to complete a screening survey. Within the online session, the moderator initiated proceedings by providing a live demonstration of *ConceptSlicer*'s application using an illustrative use case (involving the "cat" class). Subsequently, each participant was assigned three tasks that were designed in accordance with the design requirements outlined in Sec. 3.1:

- **T1: Slice Finding.** Examine the under-performing data slices and choose the ones that require the highest priority for optimization. This selection should be guided by both the model performance data and the metadata.
- **T2: Slice Explanation.** Craft natural language explanations for the identified data slices by chatting with ChatGPT. These descriptions are intended to aid in the analysis of failure modes. If the output is not satisfactory, participants can manually enhance the automatically generated description based on their comprehension of the chosen slice.
- **T3: Data Refinement.** Choose scenarios recommended by ChatGPT that closely correspond to the images found within the selected slice. Subsequently, enhance the training dataset by retrieving analogous data from the supplementary dataset and retraining the model accordingly.

Lastly, participants were requested to provide responses in the questionnaires and, based on their answers and exploratory interactions, potentially agree to further interviews. The complete session typically spanned a duration of 60 to 90 minutes.

5.3 Observations

5.3.1 Learning Curve. While all participants were already acquainted with machine learning models, particularly for tasks involving image processing, they faced a relatively steep learning curve during the online study session. In an effort to ease the cognitive load for the participants, we utilized the questionnaire as a platform to outline the objectives, recommended steps, and additional notes for each study task. Nevertheless, it was observed that the participants still strove to comprehend the underlying mechanisms of *ConceptSlicer*, which they perceived as crucial for their ongoing use of the system. Specifically, many participants encountered difficulty in grasping the definition of "concept" accurately before being able to effectively apply it in subsequent tasks. And such confusion regarding the concept's definition could potentially result in erroneous optimization decisions. Notably, we observed a link between participants' professional experience and the learning curve they encountered. This correlation between their background and familiarity with ML model training significantly influenced their intuition, logical reasoning, and interaction with the extensive language model. More details on this aspect will be presented in Sec. 5.3.2 and Sec. 5.4.1.

Include Present Concepts	Include Absent Concepts	Users	Number of Users
☑	☐	E2, E3, E6, E7, E8	5
☐	☑	E1, E4	2
☑	☑	E5	1

Figure 6: The three model optimization strategies identified in expert reviews.

5.3.2 Optimization Strategies. Due to their varying levels of experience in machine learning, the participants employed diverse strategies when addressing the model's failure (**T2**) and retrieving images for data augmentation (**T3**). We categorized their model optimization strategies based on the types of concepts they chose to include or exclude in the the prompt they fed into ChatGPT, leading to the identification of three typical patterns, as illustrated in Fig. 6. The majority, encompassing five out of eight participants, opted to incorporate more images that were similar to those within the current slice. These images contained the presenting concepts while excluding the absent concepts. Their rationale was grounded in the belief that the model failed because it "*doesn't perform very well on these concepts*" (E2). Consequently, they "*want the model to see more images (of the same kind)*" (E3, E7, E8) and ensure the model "*won't link (the presenting concepts and absent concepts) too much*"(E3). In contrast, E1 and E4 took a different approach by incorporating additional images that complemented those already in the slice. Their reasoning was that an excessive focus on the presenting concept wasn't necessary "*because it's already well represented*"(E4). A third strategy emerged from E5, who added more images featuring both the presenting and absent concepts. They advocated "*using some other (patterns) to dilute this kind of scenario*". These three strategies result in varying changes in model performance during the refinement stage. Section 5.4 presents the quantitative evaluation results and their corresponding analysis.

It's worth noting that, beyond the fundamental decisions of including or excluding different concept types, some participants explored *ConceptSlicer* in unexpected ways, leading to innovative solutions to enhance model optimization. Specifically, participants E1, E4, and E5 hypothesized that the model's subpar performance might be relevant to specific "*properties of the object, like color*"(as observed by E7). Therefore, they harnessed the chat and refinement functionalities of *ConceptSlicer* to indirectly search for images showcasing distinct color schemes, with the aim of obtaining images that offered "*additional segmentation attributes rather than focusing solely on the object itself*". For example, E7 prompted ChatGPT to describe scenarios featuring stark black and white colors, resulting in the intriguing description of "*a trial ride through a forested area with the white and back-clad rider admiring the changing autumn colors*". This description guided CLIP to retrieve the image that aligned with the participant's request. E4, on the other hand, speculated that the model's performance might be influenced by image quality. To address this, they adopted a proactive approach by excluding images with excessively small objects during the image selection

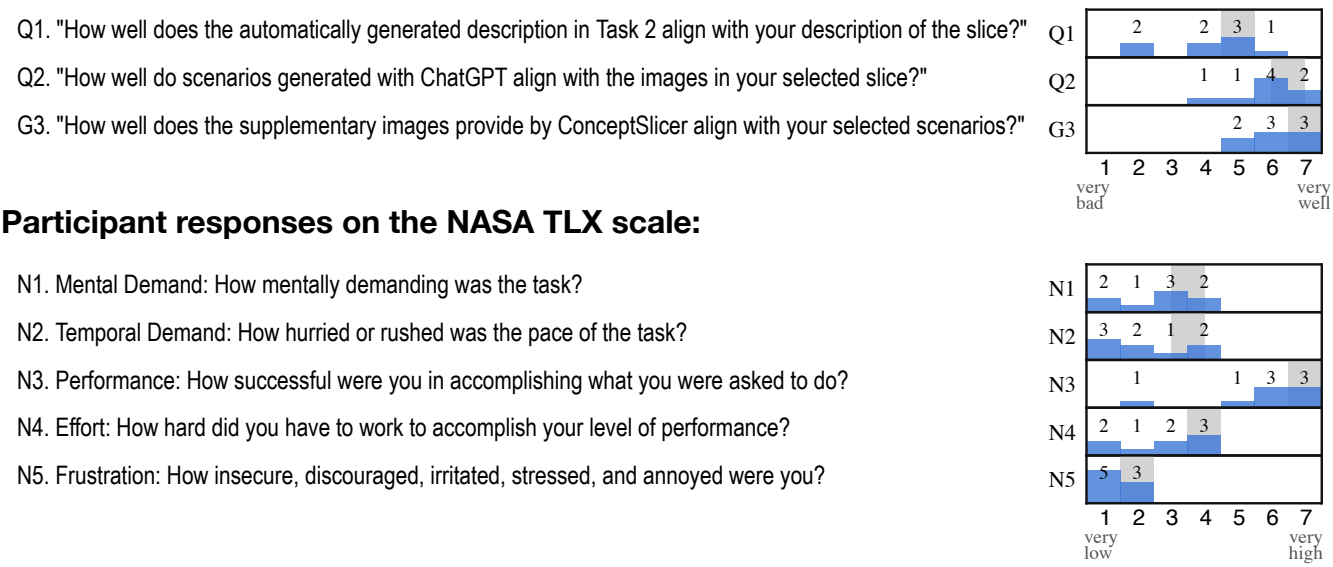


Figure 7: The participated experts and ML practitioners are generally satisfied with the result produced by *ConceptSlicer*.

process. These innovative interactions with *ConceptSlicer* provided valuable insights for the future development of the system, particularly in terms of enhancing customization support to accommodate users with diverse needs and preferences.

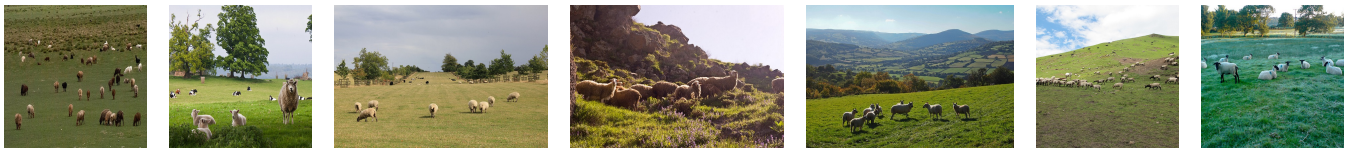
5.4 Model Optimization Results

5.4.1 Attitude towards Large Foundation Models. When building *ConceptSlicer*, we integrated the state-of-the-art foundation models to power the functionality, including a self-supervised model for visual concept detection (Sec. 3.3), ChatGPT for slice explanation (Sec. 3.4) and CLIP for supplementary image retrieval (Sec. 3.5). We identified three types attitudes towards the participants' attitude towards the functionalities powered by these models. It is worth mentioning that each participant could show multiple attitudes on different stages or components of *ConceptSlicer*.

- **Curious.** When engaging with *ConceptSlicer*, the majority of participants demonstrated a curious, and at times, a skeptical approach towards the capabilities facilitated by the foundational models. E1, E2 and E5 demonstrated their curiosity by actively posing a series of questions to the moderator, delving deep into the technical intricacies and implementation mechanisms behind *ConceptSlicer*. Conversely, E1, E4, E5, and E7 took a more playful approach, experimenting with ChatGPT through unconventional commands. For instance, they asked it to generate sentences with specific lengths and levels of detail or prompted it to describe scenarios with specific color schemes, as mentioned in Sec. 5.3.2. The overarching disposition of curiosity often transitioned into either a positive or negative stance towards the entire system, depending on the outcomes. It is also noteworthy that a substantial number of participants developed a strong interest in the potential of such extensive foundational models following their participation in the study.
- **Positive.** Three out of the eight participants (E2, E6, E8) commended the overall quality of both the prompt and the retrieved supplementary images. In particular, E8 mentioned that "*I think the the prompts actually are good...The first three are relevant*". And E6 commented that "*most (retrieved images) looks good*" and felt " *satisfied with the rest of the images*". It is noteworthy that all three participants had extensive experience in machine learning, with over five years of involvement in related topics and rated their familiarity with large foundational models as 5 out of 7. Interestingly, our observations suggest that participants who are either highly familiar (rating above 6 out of 7) or significantly less familiar (rating below 3 out of 7) with large foundational models tend to exhibit lower levels of satisfaction.
- **Negative.** Four participants expressed dissatisfaction with the outcomes generated by the foundational models. Some participants found the scenario descriptions produced by ChatGPT to be "*too abstract*" (P2) or "*too complicated*"(P4). They stated that they simply wanted concise descriptions but felt that the model was generating narrative-like content for them. Additionally, some of these participants recommended the inclusion of further illustrations or supplementary functions to facilitate a more efficient understanding of the scenarios generated by ChatGPT.

After the study session, we enhanced the training dataset by separately incorporating additional images chosen by every participant, and fine-tuned the object detection model subsequently. Notably, even though each model underwent only one epoch of fine-tuning with a relatively small number of supplementary images, we observed significant improvements in the performance of 6 out of 8 models (refer to Table 2). Meanwhile, we identified a noteworthy correlation between the outcomes of model optimization and the strategies employed by the participants, as outlined in Sec. 5.3.2. Specifically, all five participants who opted to augment the dataset

(a) Sample supplementary images selected by E1



(b) Sample supplementary images selected by E4

Figure 8: Both E1 and E4 included more images featuring absent concepts (grass) while excluding the present concept (human riding a horse). Additionally, E4 introduced a relatively large number of images featuring small-sized objects, inadvertently introducing an additional confounder (sheep), which could be a potential cause for the declined model performance.

with images containing the present concept but excluding the absent concept of the current slice succeeded in enhancing model performance on both the targeted slice and the entire dataset. In contrast, the participants who adopted the second strategies (E1 and E4) failed to improve the overall model performance, although E4 did manage to enhance performance on the targeted slice. We inspected the supplementary images chosen by both participants and found both of them added more images featuring absent concepts (grass) while excluding the present concept (human riding a horse) as illustrated in Fig. 8. This action consequently reinforced spurious correlations between grass and the class "horse". Additionally, E4 introduced a relatively large number of images featuring small-sized sheep to help the model "*differentiate what is sheep, what is horse*". While this strategy marginally improved the model's performance on this specific subset, it also inadvertently introduced an additional confounder, resulting in a decrease in overall model performance. The third strategy from E5 was also successful, but we attribute it to a pure increase in training samples. These findings suggest that augmenting the training dataset with images featuring the present concept but without the absent concept of the current slice serves as an effective approach to addressing model failures caused by spurious correlations. However, we acknowledge that the samples from the other two strategies are limited, indicating the need for further experiments in future work to substantiate their shortcomings.

6 DISCUSSIONS

Large Foundational Models. During our communication with domain experts and ML practitioners, we discovered that the quality of scenarios generated by ChatGPT and the images retrieved with CLIP is highly dependent on the specific prompt used. In fact, even a single word change in the prompt can yield significantly different results, highlighting the importance of prompt engineering. For instance, the inclusion or exclusion of the word "*briefly*" in the prompt for use scenario 1 (detailed in Sec. 4.1) can remarkably change the sentence length and level of detail in the generated scenario descriptions. Such observation aligns with many existing research that emphasize the critical role of prompt engineering [17, 59, 68]. We believe that facilitating efficient support for prompt engineering is both essential and promising for future work in this domain. Notably, various approaches have been explored to address this challenge, such as chain-of-thought prompting [57, 58, 64], least-to-most prompting [67], and human-in-the-loop solutions [37]. Meanwhile, it's worth noting that results from ChatGPT may vary even with the same prompt, hampering reproducibility. In *ConceptSlicer*, we make a trade-off by prioritizing flexibility over repeatability, enabling users to query ChatGPT multiple times until a satisfactory result is obtained. We plan to explore alternative methods in the future by enhancing the retrieval stability.

System Latency. The integration of large language models into *ConceptSlicer* introduces inherent latency when users interact with the system. This latency primarily arises from two sources: the network connection with the ChatGPT API and the query/retrieval of images from a large supplementary dataset. During our observations, we noted that some domain experts and ML practitioners adjusted their interaction strategies with *ConceptSlicer* upon realizing the presence of such latency. For instance, they tended to select a scenario description and "*quickly get some previews*" (E7) multiple times before making their final selections for image retrieval. Additionally, E4 suggested providing progress hints to help users decide whether to wait until the image retrieval process is complete. To address the latency issue, we plan to update the system to use local and more lightweight foundational models. Furthermore, we intend to implement visual hints, as suggested by the study participants, to improve the user experience and mitigate the effects of latency.

Workflow Generalizability. *ConceptSlicer* manages multiple same-class objects by utilizing the worst prediction accuracy among them in the performance data for slice finding. Although our solution is crafted to validate ML models trained for one-class classification tasks, users can employ it to address multi-class issues by diagnosing one class at a time. The methodology is easily extendable to other datasets as long as the model prediction results are available and the performance data is correctly formatted. The other required piece of information for slice finding—metadata—can be generated automatically with semantic segmentation methods and is solely related to individual image content. In our future work, we aim to facilitate users to define additional metadata, such as spatial

Table 2: Model Performance (Accuracy) for "Horse" Class. The optimized slices per user are highlighted in bold. With the exception of E1, all other users were successful in improving the performance of the data slices they chose to optimize.

	Baseline	E1	E2	E3	E4	E5	E6	E7	E8
All	0.4718	0.441	0.5026	0.5357	0.4244	0.5026	0.5178	0.5102	0.5103
Slice 1	0.34	**0.32**	0.4059	0.4356	**0.3689**	**0.38**	0.4257	0.3861	0.36
Slice 2	0.3529	0.3235	**0.4078**	**0.4369**	0.3714	0.3824	0.4272	**0.3981**	**0.3627**
Slice 3	0.3529	0.3491	0.4393	0.4673	0.3853	0.3962	0.4393	0.3925	0.3868
Slice 4	0.3774	0.3491	0.4393	0.4673	0.3853	0.3868	**0.4299**	0.4019	0.3868
Slice 5	0.3529	0.3524	0.4434	0.4717	0.4019	0.4	0.4434	0.4151	0.3905
Slice 6	0.3832	0.3458	0.4352	0.463	0.3818	0.3925	0.4352	0.3981	0.3832
Slice 7	0.3832	0.3458	0.4352	0.463	0.3818	0.3925	0.4352	0.3981	0.3832
Slice 8	0.3529	0.3458	0.4352	0.463	0.3818	0.4019	0.4352	0.4074	0.3832
Slice 9	0.3529	0.3458	0.4352	0.463	0.3818	0.3925	0.4352	0.3981	0.3832
Slice 10	0.3529	0.3458	0.4352	0.462	0.3818	0.3925	0.4352	0.3981	0.3832
Slice 11	0.3529	0.3458	0.4352	0.4722	0.3818	0.4019	0.4444	0.4074	0.3925

information [26], to provide a deeper context and more coherent explanations.

Target User Segmentation. When asked about additional functionalities they expected from *ConceptSlicer*, the experts or ML practitioners' suggestions appeared to be influenced by their experience in machine learning and their familiarity with large foundational models. In particular, their suggestions could be categorized into two main groups:

Enhancing the System for Novice Users: The study participants suggested several improvements to make the system more accessible and intuitive for those new to ML model evaluation and training.

- Thumbnail Display: One suggestion was to display thumbnails for each scenario description generated by ChatGPT. This visual aid can provide users with a quick overview of the scenario, aiding in comprehension and decision-making.
- Scenario Differentiation: Experts also recommended highlighting the differences among generated scenarios. This feature would allow users to easily compare and contrast different scenarios, making it easier to select the most appropriate one.
- Image Ranking: Another suggestion was to implement an automatic ranking system for the images retrieved by CLIP. This would streamline the selection process by presenting the most relevant images first.
- User Guidance: Some experts suggested improving the *ConceptSlicer* system by focusing on user guidance. This includes implementing intuitive navigation, providing clear instructions, and creating a responsive design that caters to various user needs and preferences.

Enhancing the System for Expert Users: The feedback from the study also highlighted the need to cater to the requirements of experienced users. They suggested several improvements to provide more control, flexibility, and detailed insights to these users.

- Model's Prediction Confidence: One suggestion was to display the model's prediction confidence for individual images. This feature would provide expert users with more detailed insights into the model's performance, enabling them to make informed decisions based on the reliability of the predictions.
- Direct Editing of the Prompt: Experts also recommended supporting direct editing of the prompt. This functionality would give experienced users greater control over the input, allowing them to tailor the prompt to their specific requirements and preferences.
- Exemplar-Based Image Query: The more experienced users have expressed the desire to have direct influence over the selection of images that are incorporated into the training dataset. For instance, E7 suggested that our system should include a feature supporting exemplar-based image queries. This feature would allow users to supply specific images to be used as references when querying the new dataset for relevant samples. Further, E7 recommended the integration of a query-by-embedding feature that would enable users to input the specific CLIP embedding they're seeking.

As part of our future work, we plan to conduct a larger-scale user survey to gather more comprehensive feedback. Based on the specific application scenario and the user group we intend to serve, we will then determine the direction in which to further develop and enhance *ConceptSlicer*.

7 CONCLUSION

In this paper, we have presented *ConceptSlicer*, a system designed to streamline the process of validating computer vision models. This is achieved through a unique approach where data slice analysis is guided by visual concepts. The *ConceptSlicer* workflow is composed of three distinct phases, each leveraging the potential of large foundational models to amplify user input and retrieve the necessary data. To bring this workflow to fruition, we have constructed an interactive system, incorporating the principles of user-friendly design and responsiveness. In order to gauge the efficiency and performance of *ConceptSlicer*, we engaged with eight domain experts or ML practitioners through interviews. The results obtained from these interviews not only emphasized the effectiveness of our system in the sphere of model validation and evaluation, but also shed a light into a range of solutions and viewpoints regarding the incorporation of large language models in executing their tasks. In conclusion, we reflect on these findings, discussing the possibilities

of integrating language models within intelligent user interfaces. Such a discourse provides insights that could shape future pursuits in this domain. The potential of systems like *ConceptSlicer* to harness the power of language models within its framework showcases a promising future for machine learning model validation.

REFERENCES

[1] Yongsu Ahn, Yu-Ru Lin, Panpan Xu, and Zeng Dai. 2023. ESCAPE: Countering Systematic Errors from Machine's Blind Spots via Interactive Visual Analysis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–16.

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403* 5, CSCW2 (2023).

[3] Guy Barash, Eitan Farchi, Ilan Jayaraman, Orna Raz, Rachel Tzoref-Brill, and Marcel Zalmanovici. 2019. Bridging the gap between ML solutions and their business requirements using feature interactions. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM, New York, NY, USA, 1048–1058.

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 6541–6549.

[5] Neslihan Bayramoglu, Juho Kannala, and Janne Heikkilä. 2016. Deep learning for magnification independent breast cancer histopathology image classification. In *2016 23rd International conference on pattern recognition (ICPR)*. IEEE, IEEE, Piscataway, NJ, 2440–2445.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Ángel Alexander Cabrera, Abraham J Druck, Jason I Hong, and Adam Perer. 2021. Discovering and validating ai errors with crowdsourced failure reports. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–22.

[8] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 11621–11631.

[9] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems* 32 (2019).

[10] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*. IEEE, Piscataway, NJ, 2722–2730.

[11] Mayee Chen, Karan Goel, Nimit S Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Ré. 2021. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*. PMLR, PMLR, London, UK, 1617–1629.

[12] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. 2018. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 7976–7985.

[13] Zhi Chen, Yijie Bei, and Cynthia Rudin. 2020. Concept whitening for interpretable image recognition. *Nature Machine Intelligence* 2, 12 (2020), 772–782.

[14] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. 2019. Slice finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, IEEE, Piscataway, NJ, 1550–1553.

[15] Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu. 2022. Zero-shot out-of-distribution detection based on the pre-trained model clip. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. AAAI, Washington, DC, USA, 6568–6576.

[16] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision* 111 (2015), 98–136.

[17] Yingchaojie Feng, Xingbo Wang, Kam Kwai Wong, Sijia Wang, Yuhong Lu, Minfeng Zhu, Baicheng Wang, and Wei Chen. 2023. PromptMagician: Interactive Prompt Engineering for Text-to-Image Creation. *arXiv preprint arXiv:2307.09036* 1 (2023).

[18] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, PMLR, London, UK, 1180–1189.

[19] Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April* 1 (2023).

[20] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards Automatic Concept-based Explanations. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., Red Hook, NY, USA. https://proceedings.neurips.cc/paper_files/paper/2019/file/77d2afcb31f6493e350fca61764efb9a-Paper.pdf

[21] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards Automatic Concept-based Explanations. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., San Diego, CA, USA. https://proceedings.neurips.cc/paper_files/paper/2019/file/77d2afcb31f6493e350fca61764efb9a-Paper.pdf

[22] Liang Gou, Lincan Zou, Nanxiang Li, Michael Hofmann, Arvind Kumar Shekar, Axel Wendt, and Liu Ren. 2020. VATLD: a visual analytics system to assess, understand and improve traffic light detection. *IEEE transactions on visualization and computer graphics* 27, 2 (2020), 261–271.

[23] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. 2016. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE transactions on medical imaging* 35, 5 (2016), 1153–1159.

[24] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921* 1 (2021).

[25] Wenbin He, William Surmeier, Arvind Kumar Shekar, Liang Gou, and Liu Ren. 2022. Self-supervised semantic segmentation grounded in visual concepts. *arXiv preprint arXiv:2203.13868* 1 (2022).

[26] Md Naimul Hoque, Wenbin He, Arvind Kumar Shekar, Liang Gou, and Liu Ren. 2022. Visual Concept Programming: A Visual Analytics Approach to Injecting Human Intelligence at Scale. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 74–83.

[27] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. 2018. Does distributionally robust supervised learning give robust classifiers?. In *International Conference on Machine Learning*. PMLR, PMLR, London, UK, 2029–2037.

[28] Jinbin Huang, Aditi Mishra, Bum Chul Kwon, and Chris Bryan. 2022. ConceptExplainer: Interactive explanation for deep neural networks from a concept perspective. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 831–841.

[29] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. 2020. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, Springer, New York, NY, USA, 124–140.

[30] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. OpenCLIP. https://doi.org/10.5281/zenodo.5143773 If you use this software, please cite it as below..

[31] Ajay Jain, Matthew Tancik, and Pieter Abbeel. 2021. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Piscataway, NJ, 5885–5894.

[32] 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.

[33] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*. PMLR, PMLR, London, UK, 2668–2677.

[34] Bum Chul Kwon, Jungsoo Lee, Chaeyeon Chung, Nyoungwoo Lee, Ho-Jin Choi, and Jaegul Choo. 2022. DASH: Visual Analytics for Debiasing Image Classification via User-Driven Synthetic Data Augmentation. *arXiv preprint arXiv:2209.06357* 1 (2022).

[35] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications* 10, 1 (2019), 1096.

[36] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. 2021. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*. PMLR, PMLR, London, UK, 7721–7735.

[37] Aditi Mishra, Utkarsh Soni, Anjana Arunkumar, Jinbin Huang, Bum Chul Kwon, and Chris Bryan. 2023. PromptAid: Prompt Exploration, Perturbation, Testing and Iteration using Visual Analytics for Large Language Models. *arXiv preprint arXiv:2304.01964* 1 (2023).

[38] Pim Moeskops, Max A Viergever, Adriënne M Mendrik, Linda S De Vries, Manon JNL Benders, and Ivana Išgum. 2016. Automatic segmentation of MR brain images with a convolutional neural network. *IEEE transactions on medical imaging* 35, 5 (2016), 1252–1261.

[39] David Munechika, Zijie J Wang, Jack Reidy, Josh Rubin, Krishna Gade, Krishnaram Kenthapadi, and Duen Horng Chau. 2022. Visual Auditor: Interactive Visualization for Detection and Summarization of Model Biases. In *2022 IEEE Visualization and Visual Analytics (VIS)*. IEEE, IEEE, Piscataway, NJ, 45–49.

[40] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. 2020. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775* 1 (2020).

[41] OpenAI. 2021. GPT-3.5. https://www.openai.com/ Accessed: September 12, 2023.

[42] OpenAI. 2022. *ChatGPT*. OpenAI. https://chat.openai.com/ Accessed: September 12, 2023.

[43] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[44] Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. *arXiv preprint arXiv:1909.02060* 1 (2019).

[45] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence. In *Proceedings of the 2021 International Conference on Management of Data*. Association for Computing Machinery, New York, NY, United States, 1400–1412.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. https://doi.org/10.48550/ARXIV.2103.00020

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, London, UK, 8748–8763. https://proceedings.mlr.press/v139/radford21a.html

[48] Svetlana Sagadeeva and Matthias Boehm. 2021. Sliceline: Fast, linear-algebra-based slice finding for ml model debugging. In *Proceedings of the 2021 International Conference on Management of Data*. Association for Computing Machinery, New York, NY, United States, 2290–2299.

[49] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and HV Jagadish. 2022. A Survey on Techniques for Identifying and Resolving Representation Bias in Data. *arXiv preprint arXiv:2203.11852* 1 (2022).

[50] Eric Slyman, Minsuk Kahng, and Stefan Lee. 2023. VLSlice: Interactive Vision-and-Language Slice Discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Piscataway, NJ, 15291–15301.

[51] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. 2020. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems* 33 (2020), 19339–19352.

[52] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html* 3, 6 (2023), 7.

[53] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* 1 (2023).

[54] Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. 2019. Credit card fraud detection-machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, IEEE, Piscataway, NJ, 1–5.

[55] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. 2022. CLIPasso: Semantically-Aware Object Sketching. *ACM TOG* 41, 4 (2022), 86:1–86:11.

[56] Xumeng Wang, Wei Chen, Jiazhi Xia, Zexian Chen, Dongshi Xu, Xiangyang Wu, Mingliang Xu, and Tobias Schreck. 2020. ConceptExplorer: Visual analysis of concept drifts in multi-source time-series data. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, IEEE, Piscataway, NJ, 1–11.

[57] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* 1 (2022).

[58] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[59] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* 1 (2023).

[60] Chin-Chia Michael Yeh, Zhongfang Zhuang, Junpeng Wang, Yan Zheng, Javid Ebrahimi, Ryan Mercer, Liang Wang, and Wei Zhang. 2021. Online Multi-horizon Transaction Metric Estimation with Multi-modal Learning in Payment Networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 4331–4340.

[61] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*. IEEE, Piscataway, NJ, 6023–6032.

[62] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*. PMLR, PMLR, London, UK, 12310–12320.

[63] Xiaoyu Zhang, Jorge Piazentin Ono, Huan Song, Liang Gou, Kwan-Liu Ma, and Liu Ren. 2022. SliceTeller: A Data Slice-Driven Approach for Machine Learning Model Validation. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 842–852.

[64] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493* 1 (2022).

[65] Zhenge Zhao, Panpan Xu, Carlos Scheidegger, and Liu Ren. 2021. Human-in-the-loop extraction of interpretable concepts in deep learning models. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 780–790.

[66] Chong Zhou, Chen Change Loy, and Bo Dai. 2022. Extract Free Dense Labels from CLIP. In *ECCV*. Springer Nature, Switzerland, 696–712.

[67] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625* 1 (2022).

[68] Haoquan Zhou and Jingbo Li. 2023. A Case Study on Scaffolding Exploratory Data Analysis for AI Pair Programmers. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–7.