How Multi-Modal LLMs Reshape Visual Deep Learning Testing? A Comprehensive Study Through the Lens of Image Mutation

LIWEN WANG, The Hong Kong University of Science and Technology, China
YUANYUAN YUAN*, Tsinghua University, China
AO SUN, The Hong Kong University of Science and Technology, China
ZONGJIE LI, The Hong Kong University of Science and Technology, China
PINGCHUAN MA, The Hong Kong University of Science and Technology, China
DAOYUAN WU†, Lingnan University, China
SHUAI WANG*, The Hong Kong University of Science and Technology, China

Visual deep learning (VDL) systems power safety-critical applications like autonomous driving and demand comprehensive testing, where generating diverse image mutations is crucial. Recently, multi-modal large language models (MLLMs) introduced instruction-driven methods, offering a unified paradigm of creating diverse and complex image mutations by describing them in text. Despite the promise, their mutation quality and applicability remain largely unexplored. We present the first study assessing MLLM's adequacy in VDL testing, focusing on the semantic *validity* of mutated images, their *alignment* with text instructions (prompts), and the *faithfulness* in preserving unchanged semantics. Through large-scale human studies and quantitative analysis, we identify MLLMs' potential. Notably, while SoTA MLLMs (e.g., GPT-4V) struggle with editing already-appeared semantics in images (e.g., rotating objects), they enable novel "semantic-replacement" mutations (e.g., *"dress a dog with clothes"*) that were infeasible previously. These mutations bring new semantics, effectively triggering unique VDL faults. Thus, MLLM-based mutations are a vital complement to traditional methods, not a replacement. Building on our findings, we develop new testing metrics tailored to MLLM-enabled mutations and summarize lessons for employing and designing MLLM mutation pipelines in different testing scenarios. Our evaluation framework and generated dataset form a reusable benchmark for future research.

CCS Concepts: • **Software and its engineering** → *Software notations and tools*; • **Computing methodologies** → *Artificial intelligence*;

Additional Key Words and Phrases: Deep Learning Testing, metamorphic testing, multi-modal large language models

*Corresponding author
†Participated in this work when working at HKUST.

Authors' Contact Information: Liwen Wang, The Hong Kong University of Science and Technology, Hong Kong, China, lwanged@cse.ust.hk; Yuanyuan Yuan, Tsinghua University, Beijing, China, yyyuan@tsinghua.edu.cn; Ao Sun, The Hong Kong University of Science and Technology, Hong Kong, China, asunac@cse.ust.hk; Zongjie Li, The Hong Kong University of Science and Technology, Hong Kong, China, zligo@cse.ust.hk; Pingchuan Ma, The Hong Kong University of Science and Technology, Hong Kong, China, pmaab@cse.ust.hk; Daoyuan Wu, Lingnan University, Hong Kong, China, daoyuanwu@ln.edu.hk; Shuai Wang, The Hong Kong University of Science and Technology, Hong Kong, China, shuaiw@cse.ust.hk.

ACM 1557-7392/2026/5-ART
https://doi.org/10.1145/3815578

1 Introduction

Visual deep learning (VDL) systems comprehend the semantics of image contents to enable a wide range of real-world applications (e.g., autonomous driving, medical imaging diagnosis) that were previously human-reliant. However, VDL systems can experience drastic failures in their outputs due to their obscure decision rules, jeopardizing the reliability of VDL systems in safety-critical applications. Software testing has been very effective in detecting VDL failures; it mutates input images and defines the accompanying testing oracle and numeric validation metric. With the metric validating the applied mutation, a VDL failure can be detected as a violation of the oracle. In recent years, our community has proposed a variety of mutation schemes for VDL testing under different scenarios, such as image classification [77], object detection [72], image captioning [80], and autonomous driving [57, 69, 86].

Inputs of VDL systems are images whose contents have rich semantics. Thus, to evaluate the reliability of VDL systems via software testing, comprehensively covering different image semantics is demanding for the input mutation. Nevertheless, mutating image semantics is fundamentally challenging because the relation between pixel values and most image semantics is unclear and hard to formulate [28, 40, 47, 65, 83, 90]. Most existing methods primarily mutate some pixel-level semantics such as brightness and contrast [57]. Some recent works support mutating more complex semantics via representation learning techniques [65, 86, 89], which however require paired images containing the expected mutations and only apply to specific domains such as driving scenes [86] or face photos [65].

The recent multi-modal large language models (MLLMs)[1], e.g., GPT-4V [6], have liberated the freedom of mutations for image semantics: users can freely describe the anticipated mutation in natural language and let MLLMs generate the mutated images. This new *instruction-driven* mutation paradigm is shedding light on (1) supporting different traditional mutations in a unified prompt-driven manner, and therefore, one may expect to easily understand, maintain, extend, and disseminate various input mutation methods in natural language texts. (2) Moreover, it also shows promising potential in offering a wide range of new mutations that are hardly achievable by traditional methods, based on the high natural language comprehension ability of LLMs, and high image synthesis capabilities of advanced vision backends (e.g., Diffusion model [31]). Thus, one may reasonably anticipate that MLLMs can revolutionize VDL testing, in parallel to large language model (LLM)'s emerging adoption and success in our community to boost traditional software testing [18, 68, 75], program repairing [22, 58, 76], and program synthesis [7, 37, 74].

Nevertheless, unlike traditional methods that explicitly form the mutations (e.g., via mathematical formulas; see Sec. 3.1), it is inherently challenging to interpret how MLLMs generate mutated outputs. Some recent reports show that MLLMs can generate semantically invalid images [13, 20, 23], which makes the follow-up testing meaningless, as those detected "failures" are induced by the VDL system's incorrect usages, not its faults. Thus, it is demanding to systematically evaluate MLLM's capability in mutating images.

Following common practice in VDL testing [69, 77, 86], we first measure the quality of MLLM-generated test inputs from: 1⟩ the *validity* of the overall semantics in mutated images, 2⟩ the *alignment* of mutated images with text instructions, and 3⟩ the *faithfulness* of how different mutations preserve semantics that should be unchanged. This is achieved via a human evaluation comprising 2,300 annotated images and 13,800 assessment questions, conducted by 20 recruited Ph.D. students experienced in VDL systems and software testing; we use Amazon Mechanical Turk [4] as a restricted-access annotation interface (i.e., not crowdsourced). Based on the quality assessment, we then study MLLM's mutation capabilities by answering two research questions (RQs). **RQ1**: Can MLLMs unify different traditional input mutations? **RQ2**: What benefits can MLLMs bring to input mutations in VDL testing? Since input validation is crucial to VDL testing, we further study if existing validation metrics

[1]Since MLLMs are often jointly used with other large models such as Diffusion model [31], CLIP [59], DALLE [3], etc., to minimize confusion, we use MLLMs as a general term for them. We use their own names when referring to their special usage.

are still applicable to MLLM-based mutations (**RQ3**). Finally, we focus on the different pipelines of employing MLLMs for image mutations and study their specialties and the performance of their ablations (**RQ4**).

We consider two recently proposed (the only two to our knowledge) pipelines of leveraging MLLMs for image mutations; both of them implement optimizations for prompts and image generation modules to provide out-of-the-box usages. Our evaluations incorporate four SoTA MLLMs (including GPT-4V) into these pipelines to implement 24 mutations across 9 mutation types, and the evaluated datasets cover general image classification [17], fine-grained dog breed identification [1], face recognition [39], and autonomous driving [16].

Our study reveals that MLLM-enabled mutations and traditional mutations are complementary in VDL testing: MLLMs cannot unify traditional mutations (which primarily edit the status of semantics that already appear in the seed images) and exhibit low faithfulness and alignment when implementing them, e.g., none of our evaluated SoTA MLLMs can achieve image rotation. However, MLLMs bring a new dimension to image mutation: they perform impressively well on "semantic-replacement" mutations that replace semantics in the seed images (e.g., "*change the background to a library*"), which are uniquely enabled by MLLMs. Besides, our results show that the two mutation pipelines of MLLMs may fit distinct scenarios, and call for careful calibrations when using them. Moreover, we find that existing validation metrics are less applicable to the new MLLM-enabled mutations due to the diversified mutants and finer-grained operations, and accordingly propose new metrics tailored to them. We also summarize lessons for using and designing pipelines of MLLM-based mutations conditioned on different testing scenarios and establish a benchmark for assessing future MLLM-based mutation using our evaluation framework and the generated dataset. In sum, this paper makes the following key contributions:

- This paper conducts the first study on the quality of test inputs generated by MLLM for VDL testing, and designs large-scale human and quantitative evaluations from the validity, alignment, and faithfulness perspectives.
- Our findings better position MLLMs and their enabled instruction-driven mutations in VDL testing. While MLLMs cannot edit existing semantics in images well like traditional mutations, they enable replacing semantics with new ones, being complementary to traditional mutations and expanding VDL testing's scope.
- Our results reveal the limitations of existing validation metrics in handling MLLM-based mutations and the distinct specialties of different optimization pipelines that employ MLLMs for implementing mutations. We accordingly propose new metrics tailored to MLLM-based mutations and translate our observations into actionable guidance for practitioners.
- We establish a reusable benchmark that enables systematic comparison of mutation pipelines, validation metrics, and testing strategies under a common setup. The benchmark includes our evaluation framework, improved metrics, baselines, and generated datasets, facilitating future research on MLLM-based VDL testing.

Paper Organization. Sec. 2 introduces background on VDL testing. Sec. 3 surveys existing mutation schemes and MLLM mutation pipelines. Sec. 4 presents research motivations and questions. Sec. 5 and Sec. 6 detail the experimental setup and human study protocol. Sec. 7 presents findings for all four RQs. Sec. 8 distills actionable insights for practitioners. Sec. 9 reflects on lessons learned, discusses technical directions, and addresses threats to validity. Sec. 10 concludes the paper.

Research Artifact. The code, data, mutation samples, and materials in our human study are provided at: https://sites.google.com/view/llm-testing-benchmark [5].

2 Background: VDL Testing

In general, a visual deep learning (VDL) system typically consists of a visual understanding module and a decision module. The visual understanding module takes an image as input and extracts features from its content. The decision module yields outputs (according to the specific VDL application domain) based on the extracted features. From this perspective, image classifiers, object detectors, image captioners, and auto-driving systems, etc., that were widely tested in our community can all be counted as VDL systems [57, 69, 72, 77, 80].

The vanilla way of evaluating VDL systems is comparing their outputs with human annotations over a set of test inputs (i.e., labeled images in standard test datasets). However, manually annotating test inputs is labor-intensive, impeding automated and large-scale evaluations. Software testing has achieved remarkable success in evaluating the reliability of VDL systems. It uses different input mutation schemes to generate diverse test inputs and forms oracles to enable automatic "annotations" for the test inputs. For example, a slightly rescaled image (e.g., no less than 50%, as determined by the validation metric) should have the same output as the original image when fed into an object detector. This way, software testing can identify VDL faults as violations of testing oracles, where human annotations are no longer required.

Test Input Generation and Validation. Since VDL processes the semantics of image contents, various mutations and validation metrics have been proposed to generate legitimate test inputs (see details in Sec. 3). In short, the mutations broadly modify different semantics to explore potential application scenarios of VDL systems. The numeric validation metrics quantify the extents of achieved mutations and the validity of test inputs, in order to control the mutation and ensure meaningful testing results. Note that different from adversarial perturbations[2] that add invisible noise to images and study the "worst-case" attack surfaces of VDL systems, mutations employed in VDL testing holistically alter images to diversify image semantics, exploring different usage scenarios of VDL systems.

Testing Oracle. Metamorphic testing (MT) stands out as one mainstream testing paradigm of VDL testing in recent years [69, 72, 77, 80, 83, 84, 86]. MT checks whether a VDL system's outputs are aligned with the relation (e.g., an equality relation) defined in an MT testing oracle, when the VDL is processing an input and its mutated version. Unalignment shows that one of the inputs is ill-processed and a VDL fault is triggered. Since testing oracles are critical to detect VDL faults, input mutations themselves should properly preserve the oracles (e.g., within specific semantic ranges, as determined by the validation metric); see examples in Sec. 3.

3 Image Mutations and Validation

Image Decomposition: A Motivating Example. To understand VDL and the design of input mutations, we first decompose the dog image in Fig. 1 into different semantics. Starting from a lower level, the image is a matrix of pixels, and the magnitude of pixel values determines the brightness and contrast of the image. Besides, the spatial arrangement of pixels reflects geometrical properties of the image, such as the scale and the rotation angle. A reliable VDL system that recognizes dogs should be resilient to changes of these semantics. At a higher level, the image contains a dog and the background grass; the dog itself has rich semantics such as the action and the orientation. Similarly, the semantics in the background may vary with the working scenario of the VDL, e.g., the grass may be replaced by snow in a winter scene. Overall, the dog-recognition VDL should focus on the semantics of the dog and be invariant to irrelevant semantics in the background.

Based on the above semantic decomposition, the following image mutation schemes have been proposed.

3.1 Explicit & Mathematical Transformations

Pixel-level. Mutations belonging to this category can be formed as uniformly changing pixel values via $\mathbf{x}$ op c, where c is a scalar value and $\mathbf{x}$ is a matrix denoting the seed image. op decides the mutated semantics. For

[2]Such perturbations are leveraged by adversarial attacks to generate malicious inputs (i.e., adversarial examples) to fool a VDL system [26].

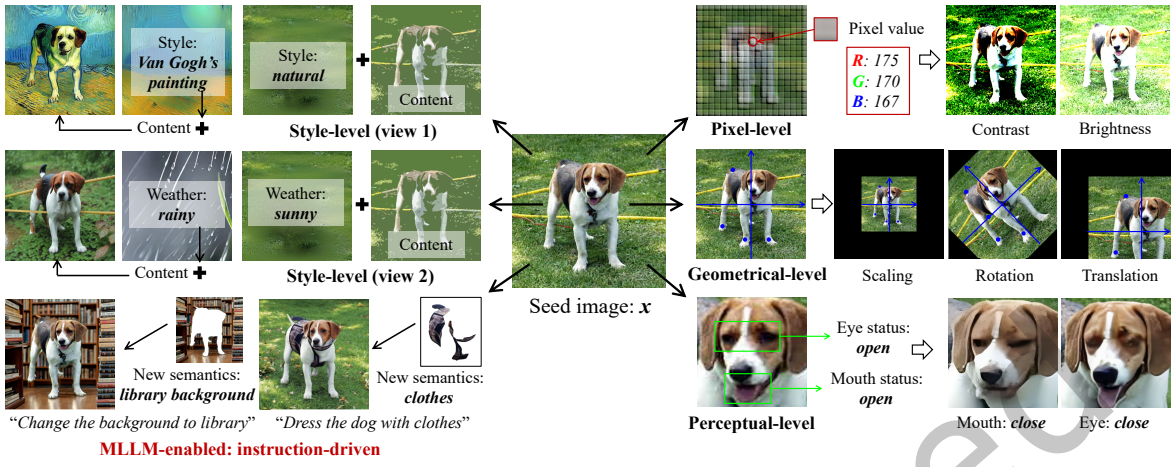


Fig. 1. Decomposition of image semantics and their corresponding mutation schemes.

example, op is the add operation if the mutation changes the image brightness, i.e., adding all pixel values in $\mathbf{x}$ with a positive number c increases $\mathbf{x}$'s brightness. Similarly, mutating image contrast can be formulated as $\mathbf{x} \times c$, which enlarges the difference between pixel values in $\mathbf{x}$. Representative examples are shown in Fig. 1.

Geometrical-level. As geometrical properties are determined by the spatial arrangement of pixels, existing works implement these mutations using affine transformations, which essentially move pixels to new localizations and preserve the image's parallelism (i.e., two parallel lines remain parallel in the mutated image). Formally, suppose pixels of the image $\mathbf{x}$ are in a rectangular coordinate and the centering pixel is the origin, the (i, j)-th pixel's localization in the mutated image can be calculated as $[i', j'] = \mathbf{A} \times [i, j] + \mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ characterize the geometrical mutation. For example, $(\mathbf{A}_R, \mathbf{b}_R)$ and $(\mathbf{A}_T, \mathbf{b}_T)$, w.r.t. rotation and translation mutations, are shown in Eq. 1,

$$\mathbf{A}_R = \begin{bmatrix} \cos\theta, & -\sin\theta \\ \sin\theta, & \cos\theta \end{bmatrix}, \mathbf{b}_R = \mathbf{0}; \ \mathbf{A}_T = \mathbf{0}, \mathbf{b}_T = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{1}$$

where θ specifies the rotating angle. t_x and t_y indicate the vertical and horizontal translation distances (see Fig. 1).

Validation and Application Scope. Validating the above mutations is straightforward: developers can directly control the mutations via their parameters (e.g., c in pixel-level mutations) and ensure generating valid test inputs by setting reasonable parameter values. These mutations are mostly adopted in testing safety critical VDL systems like autonomous driving [57, 69] due to their neat and explicit formulations. In addition, given these mutations' applicability to almost all RGB images, they are widely used in testing more general VDL tasks like image classification [77].

3.2 Implicit & Data Exploration

Recent works diversify the mutated semantics with data-driven mutations. These mutations implicitly explore patterns in some images and then apply the patterns to mutate others.

Style-level & Application Scope. From a high level, each image can be decomposed as its style and content, and VDL systems should primarily focus on the content rather than style. Thus, existing VDL testing works propose style-level mutations, which generate images of distinct styles but presumably retain the content of the seed image. The "style" has been studied from two aspects. First, the style can be viewed as describing how the

image is produced, e.g., the seed image in Fig. 1 is taken in reality and its style is "natural". In that sense, the style can be changed by generating an image's artistic-stylized variants (e.g., an image having the same dog but painted using a Van Gogh style, as in Fig. 1). These mutations over "artistic styles" have been shown effective in flipping image classifier's outputs in recent testing works [78, 83], and some works also employ them to reveal VDL system's bias to non-content semantics [25], showing their high value in VDL research.

Besides, some other works decompose the weather condition as the style and separate it from the main content. The weather conditions are transferred between images to generate test inputs, as shown in Fig. 1. However, these weather-targeted style-level mutations are specifically designed for testing VDL-based autonomous driving [86], and show limited applications in other testing scenarios.

Perceptual-level & Application Scope. Recent VDL testing focuses on fine-grained semantics of the objects in images and implements perceptual-level mutations [21, 38, 83]. For instance, in Fig. 1, the dog's eyes and mouth can be mutated to evaluate whether the VDL accurately recognizes different dogs [21]. Similar techniques can be applied to face photos for testing and improving finer-grained VDL tasks like face recognition [53, 70], face aging [50, 54], crowd counting [49, 71], etc.

Validation. Both style-level and perceptual-level mutations are implemented based on generative models like GANs [39, 86, 92]. To control the mutations, existing works propose optimization-based solutions, e.g., the cycle-consistency mechanism is formed during training in [86, 92] to better separate styles from images. However, such constraints derived from optimizations lack explicit guarantee, e.g., the cycle-consistency can be violated when transferring styles among images of dissimilar objects [83, 91]. Hence, existing testing works additionally adopt numeric metrics, such as Inception Score (IS) [64], Frechet Inception Distance (FID) [30], to measure the validity of generated test inputs. Since variations on fine-grained semantics in style- and perceptual-level mutations can hardly be captured via pixel values, recent works propose to measure the extents of achieved mutations via image embedding[3] [38, 55, 83] (e.g., obtaining style and content embeddings, or using embedding models like CLIP [55, 59] to compare image semantics with text descriptions).

3.3 MLLM-Based Mutations

In line with past image mutations in VDL testing, we explain how MLLMs can boost the mutation schemes from two aspects.

Unification & Generalization: Instruction-Driven. MLLMs have revolutionized image mutation by enabling text-to-image translation. Users can describe the desired mutation for a seed image using text instructions, and MLLMs generate the mutated image accordingly. This instruction-driven paradigm is supposed to enable VDL developers to uniformly implement previous mutations (Sec. 3.1 and Sec. 3.2) and apply domain-specific mutations (Sec. 3.2) to more general scenarios, by designing the proper text instruction.

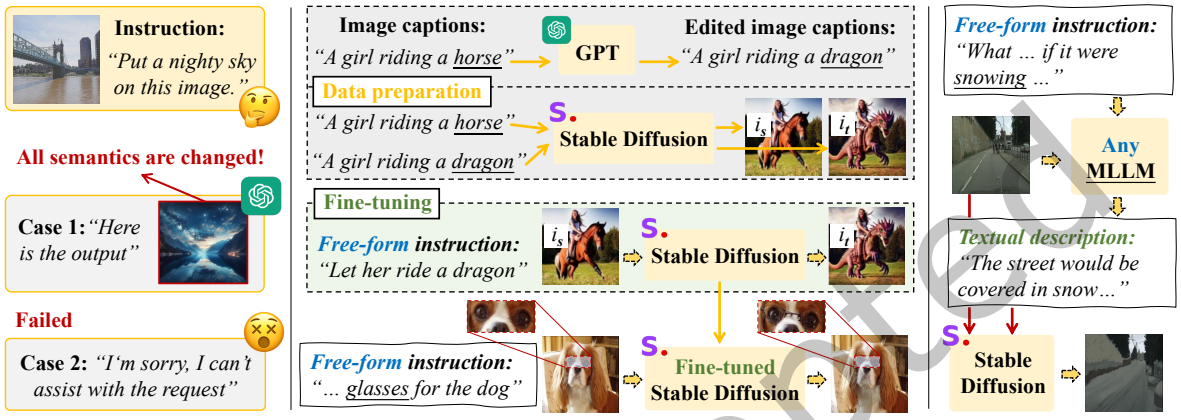
Expansion: Semantic-Replacement. The instruction-driven paradigm also largely expands the covered semantics in image mutations. Unlike previous mutations that edit the status of semantics already presented in the seed image, MLLMs bring new mutations that can replace image semantics with new ones; we refer to this new category as semantic-replacement in the rest of this paper. For example, the dog image's main body can be replaced with clothes (see Fig. 1) to further challenge the VDL system's recognition.

Comparison with Object Insertion. Knowledgeable readers may wonder about the differences between semantic-replacement mutations and the object insertion adopted in testing object-based VDL systems [72, 80, 84]. We clarify that semantic-replacement mutations operate at the semantic level and are finer-grained, which often require reasoning about the relation and hierarchy between semantics (either intra- or inter-object). For example, the "dressing with clothes" mutation in Fig. 1 needs to understand and identify the dog's body and *replace* it with appropriate clothes. Object insertion, in contrast, does not perceive image semantics and only relies on object

[3]A representation of the image that captures its semantics. The term embedding may be alternatively dubbed as "feature" in some works.

locations in the image, so that the inserted object will not overlap with existing ones. Moreover, prior object insertion mutation is not end-to-end, as they rely on object localization results from object detectors; we thus omit it in this study.

3.4 Pipelines for Implementing Mutations with MLLMs



(a) Dialog-based (b) Unidirectional (c) Bidirectional

Fig. 2. Pipelines of MLLM-based mutations. Fig. 2(a) shows the most straightforward dialog-based pipeline. Fig. 2(b)-(c) illustrates two pipelines specifically optimized for image mutations. The unidirectional pipeline shown in Fig. 2(b) requires fine-tuning a MLLM and should be used along with this tailored MLLM. The bidirectional pipeline presented in Fig. 2(c) is post-hoc and supports incorporating different MLLMs.

Dialog-Based. The most straightforward way to mutate images is building a dialog with the MLLM, as shown in Fig. 2(a). For example, the users first provide the seed image and a prompt of the mutation instruction to GPT-4V and let GPT-4V return the mutated image. However, our preliminary explorations show that this dialog-based pipeline fails to generate test inputs in *all* cases we try: GPT-4V either generates an entirely new image (nearly all semantics are changed) or responds that it cannot generate the image. Similar issues are also noted in [24, 35, 36],

To alleviate this issue, one may expect to use prompt engineering to better guide the MLLM. Nevertheless, our preliminary study shows that it is hard to automatically tune MLLM prompts and achieve improved performance constantly on different images. Moreover, we argue that "tuning prompts" is less practical in our context, as VDL developers (main users of VDL testing tools) are often not experts in prompt engineering; MLLM should provide out-of-the-box support for implementing mutations. In this regard, recent works propose to equip MLLMs with Diffusion model [61], a more sophisticated image generation model, and design two representative optimization pipelines to orchestrate their specialties in cross-modal translation and image generation [12, 14, 24, 35, 36].

Unidirectional & Fine-tuning-based. As shown in the bottom of Fig. 2(b), InstP2P designs an unidirectional pipeline which directly uses a free-form text instruction and the seed image to generate the mutated image. In general, since the text instruction primarily describes the mutation and does not include details in the images, InstP2P requires fine-tuning a Diffusion model to support diverse mutations.

Specifically, as illustrated in the top and middle of Fig. 2(b), a GPT model is first employed to edit a given textual caption to obtain a new caption. Then, the two captions are fed into a pretrained Stable Diffusion model [61] to generate two images, i_s and i_t, respectively. Finally, a free-form textual instruction is generated using GPT to

briefly summarize the changes between the two captions; this textual instruction, together with the source image i_s and target image i_t, forms a data point for fine-tuning the Diffusion model. The model is subsequently tuned to generate the target image i_t when conditioned on the source image i_s and the textual instruction. After fine-tuning, the Diffusion model is able to generate a mutated image for any unseen images and textual instructions. We refer interested readers to [11] for the fine-tuning details.

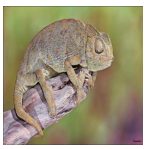
Instruction	Seed image	Textual description generated by MLLMs
"What would it look like if it were snowing?"		*"The lizard would be surrounded by a snowy environment, which would create a contrasting and unusual scene. The lizard's vibrant colors, such as the yellow and orange, would stand out against the white snow, making it visually striking. The snow would also cover the tree branch and the surrounding area, creating a wintry atmosphere. The lizard's behavior might change as it tries to adapt to the cold and snowy conditions, possibly seeking shelter or finding a way to stay warm."*
"Put on a pair of glasses for the dog."		*"It would likely appear more adorable and endearing. The glasses would add a touch of humor and personality to the dog, making it appear more human-like. The image would likely evoke a sense of amusement and warmth, as the dog's facial expression and body language would be enhanced by the glasses, creating a more engaging and entertaining visual experience for the viewer."*
"Dress the dog with clothes."		*"The scene would likely be more whimsical and amusing. The dogs, which are currently sitting on a wooden bridge or boardwalk, would appear more human-like with clothing. The clothes could be in various styles, such as coats, hats, or even costumes, depending on the occasion or the owner's preferences. The image would likely evoke a sense of humor and delight, as it is not common to see dogs dressed in clothing."*

Fig. 3. Textual descriptions generated in the bidirectional & post-hoc pipeline of MLLM-based mutations.

Bidirectional & Post-hoc. Another pipeline, as designed in [12, 14, 24, 35, 36], leverages MLLMs to transform the mutation instruction based on the seed image, such that details in the seed image can be included in the transformed instruction. As shown in Fig. 2(c), this pipeline is categorized as bidirectional, since it implements the workflow of image → text → image. Specifically, due to MLLM's promising capability of understanding images, users first feed the seed image to the MLLM and let the MLLM describe what the image would look like after applying a mutation. Here, the MLLM's output is a textual description of semantic-level details in the mutated image. This finer-grained text description, together with the seed image, is then fed to the Diffusion model to generate the mutated image. The superiority of this pipeline (compared with the dialog-based one) has been empirically justified in [14, 24, 35, 36].

Fig. 3 shows examples of textual descriptions in the bidirectional pipeline (more examples are provided in the artifact [5]). The text can precisely describe the mutation expected by the instruction. For instance, in the first case shown in Fig. 3, the description *"The lizard would be surrounded by a snowy environment..."* precisely captures the semantic transformation specified by the mutation instruction *"What would it look like if it were snowing"*, facilitating accurate execution of the intended mutation. In addition, the text includes many detailed and informative descriptions, e.g., *"The snow would also cover the tree branch and the surrounding area..."* in the first case, which gives more information of the mutated image to ease the mutant generation.

Overall, the bidirectional pipeline, to some extent, achieves automated prompt engineering by enriching the mutation instruction with the seed image's details. The unidirectional pipeline, on the other hand, fine-tunes the Diffusion model to strengthen its capability in achieving image mutations.

4 Research Motivations

Limitations of Past Methods. Although the fast development of input mutations shows promising results in VDL testing, maintaining and developing them often requires expertise in different domains. Moreover, the more advanced style- and perceptual-level mutations show limited application scopes. For instance, weather-targeted mutations may perform worse in non-driving-scene images, and the face-related perceptual-level mutations only apply to highly aligned face photos (as they rely on a unified mask to pinpoint eyes, mouth, etc.). Some perceptual-level mutations may require manually annotating the to-be-mutated semantics [65, 89].

From these perspectives, we envision that MLLMs can offer an *instruction-driven* mutation paradigm, which is more general, automated, and highly flexible for VDL testing. MLLMs have shown high capabilities in natural language understanding and image synthesis, and therefore, various kinds of mutations that were hardly or tediously achieved by traditional methods may be easily implemented via MLLMs by directly providing instructions. Moreover, since various (subtle or complex) mutations are unifiedly described in natural language (as shown in Fig. 1), the mutations can be easily understood, maintained, calibrated, and extended by VDL developers and the community.[4]

Unclear MLLM Internals. Despite the optimism, MLLM's image generation is hard to interpret due to the unclear internals. Per recent reports, MLLMs often fail to mutate images according to the given instructions [45], and may generate broken and ill-formed images from time to time [13, 20, 23]. Hence, we believe our community still lacks understanding of how exactly MLLMs can boost VDL testing, and a comprehensive study is urgent for VDL developers to understand MLLM's (in-)capability of mutating images. In addition, considering that input validation is crucial to ensure meaningful testing results, it is also demanding to evaluate whether existing validation metrics can still be applied to MLLM-based mutations.

5 Research Study Setup

Our study answers the following research questions (RQs).

- **RQ1:** *Can MLLMs unify different traditional input mutations?* We answer this RQ by assessing the quality of 1) how MLLMs re-implement traditional mutations and 2) how MLLMs extend prior domain-specific mutations to general scenarios.
- **RQ2:** *What benefits can different MLLMs bring to input mutations in VDL testing?* We answer this RQ by 1) assessing the quality of test inputs generated by the semantic-replacement mutations enabled by MLLMs, 2) evaluating their effectiveness in identifying VDL faults, and 3) analyzing the specialties of different MLLMs and their mutation pipelines.
- **RQ3:** *Are existing input validation metrics reliable when facing MLLM-based mutations, and can we develop more effective automated metrics?* We answer this RQ by evaluating whether existing validation metrics can reflect 1) the quality of test inputs generated by MLLMs, 2) the varied capabilities of different MLLMs, and 3) whether we can develop more effective automated metrics.
- **RQ4:** *What roles do different components play in different MLLM pipelines?* We answer this RQ by conducting an ablation study to disentangle the individual impacts of different components, such as the enriched prompts and the fine-tuned image generator, and analyze in depth what contributes to the distinct specialties of different MLLM pipelines.

Our RQs are based on the quality assessment of test inputs generated via different mutations, which requires perceiving and understanding image semantics. We therefore conduct a large-scale human study to evaluate image quality. Below, we introduce our evaluated aspects for the quality assessment.

[4]In some sense, this new *instruction-driven* mutation paradigm is comparable to how query-based software security testing tools, e.g., CodeQL [2], maintain varying types of vulnerabilities in a unified query language.

5.1 Evaluated Aspects

Following design principles in existing input mutations and validation metrics, we consider the following three aspects: alignment, faithfulness, and validity. These three aspects represent the essential properties of a successful mutation for VDL testing: a mutation must (1) perform the intended transformation (alignment), (2) avoid unintended side effects (faithfulness), and (3) ensure the resulting test case is plausible and realistic (validity).

Alignment. The achieved mutation must align with the user's instruction or expectation. This is critical for diagnosis; users must know which specific semantic change triggered a VDL fault. For example, if a "rotate 180 degrees" mutation fails to rotate the image, a passing test provides a false sense of security (a false negative) regarding the model's rotational invariance.

Faithfulness. A mutation must faithfully preserve all semantics that are intended to be unchanged. For example, when applying a "close the eyes" mutation to a face, the background, hair, and other facial features should remain identical. A lack of faithfulness (e.g., also changing the background) can break the testing oracle or introduce confounding variables, leading to spurious failures (false positives) and misleading diagnoses.

Validity. Finally, we holistically consider the perceptual validity of the entire mutated image. This is crucial because a mutation can be perfectly aligned and faithful, yet still produce an unrealistic image. For example, as will be shown in Fig. 4, GPT-4V replaces a portrait's eyes with closed ones (potentially from other portraits in its "database") when asked to close the eyes. Although the mutation aligns with the requirement and other semantics are untouched, the sharp transition around eyes makes the mutated image unrealistic, which rarely appears in real-world scenarios.

5.2 MLLMs and Their Pipelines

As mentioned in Sec. 3.4, two pipelines have been proposed by existing work to employ MLLMs for image mutations. They are optimized from different angles to avoid issues in prompt engineering or image generation modules, providing out-of-the-box usages for non-experts in MLLMs. In particular, the unidirectional pipeline relies on a specifically fine-tuned MLLM, we therefore use the MLLM provided by the authors [11], along with this pipeline, as a whole for evaluations and refer to it as InstP2P. The bidirectional pipeline is post-hoc and supports incorporating different MLLMs; thus, we use three SoTA and representative MLLMs, including LLaVA [48], CogVLM [73], and GPT-4V [6] into this pipeline during evaluations; the three resulting mutation methods are denoted as LLaVA, CogVLM, and GPT-4V, respectively. It's worth noting all four methods, CogVLM, LLaVA, GPT-4V, and InstP2P (before fine-tuning), use the same Stable Diffusion model [61] as the image generation module, ensuring a fair comparison for the capabilities of different MLLMs.

5.3 Datasets and Mutations

This section describes the datasets and mutations used in our evaluation. We first introduce the four datasets that cover diverse testing scenarios, and then present the mutations we consider, which are from all categories introduced in Sec. 3 and include both traditional and MLLM-enabled ones. Finally, we explain how we form two mutation sets to balance the depth and scale of our evaluation.

Datasets. Table 1 lists the evaluated datasets. They are large-scale and representative to cover different application scenarios. ImageNet consists of 1,000 classes of real-life images, and is widely used for general image classification tasks. Dog-Breed dataset contains dog images with 120 different breeds and is a subset of ImageNet. It is constructed to improve the VDL model's accuracy in domain-specific applications. FFHQ includes human face photos; it is widely used for face recognition and annotation tasks. CityScape is composed of urban street scene images, which are popular in autonomous driving, object detection, and instance segmentation, etc.

Traditional Mutations. Mutations ①-⑯ in Table 2 are traditional mutations considered by us; they cover all categories introduced in Sec. 3, including pixel-level, geometric, style, and perceptual, and each category

Table 1. **Evaluated datasets and their corresponding mutations.**

Dataset	#Classes	Description	VDL Tasks	Mutations
ImageNet [17]	1,000	Real-life images	Image Classification	①-⑦, ⑧, ⑩, ⑫, ⑬, ⑱, ⑳, ㉒, ㉔
Dog-Breed [1]	120	Dog images	Image Classification	⑰, ⑲, ㉑, ㉓
FFHQ [39]	N/A	Face photos	Face recognition, annotation, etc.	⑭, ⑮, ⑯
CityScape [16]	N/A	Urban street scene images	Autonomous driving, object detection, instance segmentation, etc.	⑨, ⑪

Table 2. **Mutations, data sources, evaluation contexts, and text instructions.**

Mutation	Type	Data	Eval	Instruction
① Contrast	Pixel	IN	H+A	"*Increase the contrast of the image.*"
② Brightness			A	"*Increase the brightness of the image.*"
③ Noise			A	"*Add random Gaussian noise to the image.*"
④ Blurring			A	"*Apply blur to the image.*"
⑤ Rotation	Geometric	IN	H+A	"*Rotate the image 180 degrees.*"
⑥ Scaling			A	"*Scale the image to 50% of its original size.*"
⑦ Translation			A	"*Translate the image by 10 pixels.*"
⑧ $\texttt{Snow}_G$	Style (Weather)	IN	H+A	"*What would it look like if it were snowing?*" (general)
⑨ $\texttt{Snow}_D$		CS	H+A	"*What would it look like if it were snowing?*" (driving)
⑩ $\texttt{Rain}_G$		IN	A	"*What would it look like if it were raining?*" (general)
⑪ $\texttt{Rain}_D$		CS	A	"*What would it look like if it were raining?*" (driving)
⑫ Monet	Style (Artistic)	IN	A	"*Change the style of this image to Monet style.*"
⑬ VanGogh			H+A	"*Change the style of this image to Van Gogh style.*"
⑭ Eyes	Perceptual	FF	H+A	"*Make the eyes of the person close.*"
⑮ Nose			A	"*Make the nose of the person short.*"
⑯ Mouth			A	"*Make the mouth of the person close.*"
⑰ Library	Background	DB	H+A	For dog images: "*Change the background to a library.*"
⑱ $\texttt{Background}_G$		IN	A	LLM generates background mutation instructions conditioned on class label.
⑲ Clothes	Subject (Major)	DB	H+A	For dog images: "*Dress the dog with clothes.*"
⑳ $\texttt{Subject}_G^{\text{Maj}}$		IN	A	LLM generates major semantics mutation instructions conditioned on class label.
㉑ Legs	Subject (Moderate)	DB	H+A	For dog images: "*Change the dog's legs to be bionic.*"
㉒ $\texttt{Subject}_G^{\text{Mod}}$		IN	A	LLM generates moderate semantics mutation instructions conditioned on class label.
㉓ Glasses	Subject (Minor)	DB	H+A	For dog images: "*Put on a pair of glasses for the dog.*"
㉔ $\texttt{Subject}_G^{\text{Min}}$		IN	A	LLM generates minor semantics mutation instructions conditioned on class label.

- **Data**: IN = ImageNet, DB = Dog-Breed, FF = FFHQ, CS = CityScape.
- **Eval**: H = human study, A = automated evaluation. "H+A" marks the **foundational set** (10 mutations: one per type, except `Style (Weather)` with two), used in both the human study (50 seeds/mutation) and automated evaluation (250 seeds/mutation). "A"-only mutations appear solely in automated evaluation.

contains multiple mutation instances for comprehensiveness. In particular, some mutations deserve special attention. First, weather transfer mutations ($\texttt{Rain}_G$, $\texttt{Rain}_D$, $\texttt{Snow}_G$, $\texttt{Snow}_D$) are context-dependent: we apply them to both general images from ImageNet and driving scenes from CityScape to capture their varying effects. Second, perceptual mutations (`Eyes`, `Nose`, `Mouth`) are currently limited to face images since they have rich fine-grained facial features; thus, these mutations can only be applied to aligned face photos from the FFHQ dataset.

MLLM-Enabled Mutations. Beyond traditional mutations, we design semantic-replacement mutations that leverage MLLMs' understanding of image content (⑰-㉔ in Table 2). These mutations operate at different semantic granularities: one targets background, while three others focus on progressively finer-grained aspects of the main subject (major, moderate, and minor features). This design enables us to assess how VDL models handle semantic changes at multiple levels. Specifically, we operationalize these three granularities as follows: a *Major* mutation

alters the subject's overall appearance (e.g., `Clothes`: dressing a dog's entire body); a *Moderate* mutation modifies a single localized body part (e.g., `Legs`: replacing legs with bionic ones); a *Minor* mutation adds a small accessory (e.g., `Glasses`: putting on glasses). We instantiate these mutations on two datasets. On Dog-Breed, we manually craft four mutation instructions (`Library`, `Clothes`, `Legs`, `Glasses`) at different scales to enable in-depth quality analysis with human verification. On ImageNet, we consider all classes and use an LLM to automatically generate class-specific instructions for four mutations ($\texttt{Background}_G$, $\texttt{Subject}_G^{\text{Maj}}$, $\texttt{Subject}_G^{\text{Mod}}$, $\texttt{Subject}_G^{\text{Min}}$) following the four granularities, which enables automated evaluation across diverse object categories to extend the analysis breadth.

Seed Image Sampling. For each mutation type shown in Table 2, we sample seed images exclusively from one dataset that is most appropriate. Once sampled, the same seed images are consistently used across all mutations within that type, mitigating potential inconsistencies arising from seed variations. This strategy also ensures that all mutations are applied in a meaningful manner and all seed images have a fair chance to be mutated. Table 1 shows the mappings between datasets and mutations. Specifically, we use CityScape for driving scene mutations (⑨, ⑪), FFHQ for face-related mutations (⑭, ⑮, ⑯), and Dog-Breed for dog-specific mutations (⑰, ⑲, ㉑, ㉓). All other mutations (which are for general images) are applied on ImageNet given its diverse object categories.

Prompt. To implement a mutation (either traditional or MLLM-enabled) using MLLMs, a text instruction (i.e., prompt) is required. We list the text instruction of each mutation in Table 2. Recall as mentioned in Sec. 3.4, the two optimized MLLM pipelines are specifically designed for implementing image mutations, so that a mutation's prompt can be a single-sentence instruction, enabling an out-of-the-box usage. Due to such a simple form, we find that the MLLM is mostly robust to changes over those instructions, and we observe that, the instruction's wording, grammatical structure, etc., does *not* notably impact the achieved mutation. Note that for Dog-Breed mutations, we manually craft instructions like "*Dress the dog with clothes.*" For ImageNet mutations, manual crafting becomes impractical due to the large number of classes (i.e., 1000). We instead use an LLM to automatically generate class-specific instructions. We constrain the LLM to follow our hand-crafted style: concise, single-sentence prompts semantically appropriate for each class. This ensures both scalability and consistency.

Two Evaluation Sets. We organize the 24 mutations into two complementary sets to balance evaluation depth and breadth. Our 24 mutations comprise 16 traditional (①–⑯) and 8 MLLM-enabled (⑰–㉔), grouped into 9 mutation types (Table 2). The first one is the *foundational set* that includes 10 representative mutations (as noted in Table 2), selecting one mutation per type except for `Style (Weather)`, which includes two representatives ($\texttt{Snow}_G$ and $\texttt{Snow}_D$) to cover both general-image and driving-scene contexts. Because human evaluation is resource-intensive, the foundational set keeps the study at a manageable scale while preserving complete type coverage. Within each type, we select the mutation with the most concrete instruction and the dataset best suited for human verification (6 traditional, 4 MLLM-enabled; marked "H+A" in Table 2). The second one is the *full set*, which includes all 24 mutations. The foundational set enables in-depth human assessment, while the full set supports large-scale automated evaluation. Automated metrics incur no per-sample human cost, so the full set scales to all mutations for broader quantitative coverage. Specifically, the human study evaluates the foundational set, whereas the automated evaluation covers the full set, which subsumes the foundational set. For the foundational set, we sample 50 seed images *per mutation*, yielding 2,300 mutated images ($6 \times 50 \times 5 + 4 \times 50 \times 4 = 2,300$) and 13,800 questions (see Sec. 6 for details). For the full set, we scale up to 250 seed images *per mutation*, producing 28,000 mutated images ($16 \times 250 \times 5 + 8 \times 250 \times 4 = 28,000$); this larger corpus enables comprehensive evaluation across diverse mutations and image categories.

6 Human Studies Setup

We form a group of 20 participants to evaluate on the *foundational set* of 10 mutations listed in Table 2. The evaluation considers the alignment, faithfulness, and validity requirements discussed in Sec. 5.1. Since evaluating

whether a mutation fulfills these requirements needs inspecting how the changed semantics (both anticipated and unanticipated) may affect the downstream VDL testing and analysis, we recruit 20 Ph.D. students experienced in software testing and VDL systems as annotators.

Question Design. The common practice, as in most existing MLLM works [66, 87], is letting humans rank different mutated images. However, we observed that MLLMs can fail in implementing a mutation or generating valid images, which cannot be reflected in the ranking results. To address this, we design the questions as follows. Each question in our human study is formed by a seed image and one of its mutated variants (using different mutations), and a participant is asked to give a score (0 to 5) for the *alignment*, *faithfulness*, and *validity* of a mutated image. A zero score indicates that the mutated image fails to satisfy one requirement. For example, if the rotation mutation implemented using a MLLM does not rotate an image, the alignment score should be zero. We use scores 1-5 to assess each requirement because one mutation has a maximum of five different implementations (i.e., one traditional method + four MLLMs); the five different choices should be distinguishable for different implementations.

Question Assignment. We let each participant evaluate one of the ten mutations, as interchangeably evaluating different mutations may confuse the participants, especially for subtle mutations that require careful comparison. Thus, each participant evaluates 200 (for MLLM-enabled mutations) or 250 (for traditional mutations) mutated images. The orders of the questions are shuffled to avoid potential bias due to question orders. To reduce personal preferences, each mutation is assigned to two different participants and their results are averaged to obtain the final score for each mutated image. We also adopt Kappa statistics to measure the inter-annotator agreement.

It's worth mentioning that the entire evaluation is conducted in a blind manner: for every mutated image, participants are not informed which mutation implementation (traditional method or one of the four MLLMs) is used to generate it. This design prevents confirmation bias and ensures that ratings reflect the intrinsic quality of mutated images rather than preconceptions about specific generation methods.

Training Session. Before starting the human study, we prepare a 30-minute teaching session for each participant. This session detailed our evaluation criteria (i.e., alignment, faithfulness, and validity) using concrete definitions, illustrative examples, and a practice round. For full transparency, the complete training protocol is included in our public research artifact [5]. Our human study uses Amazon Mechanical Turk [4] as a restricted-access annotation interface for our recruited participants (i.e., not crowdsourced). We do not set a time limit for each question because we expect participants to carefully assess the mutated images. The participant can pause and resume the evaluation at any time to avoid fatigue. The average time to finish all evaluation questions (i.e., 200 or 250) is around two hours as reported by participants.

Sanity Check. To rule out invalid evaluations (e.g., a participant misunderstands the requirement), following previous work [81], we insert sanity-check questions into each evaluation without notifying the participants. Specifically, we insert 10 questions (in a blind manner) where the "mutated" image is identical to the seed image, and the participant is expected to assign 0, 5, 5 scores for the alignment, faithfulness, and validity requirements, respectively. If a participant fails to pass 80% of the sanity-check questions, we will discard all his/her answered questions.

7 Results and Findings

This section presents the results of our human study and answers the four RQs mentioned in Sec. 5. Each subsection addresses one RQ independently and concludes with a summary box: Sec. 7.1 (RQ1: unifying traditional mutations), Sec. 7.2 (RQ2: new MLLM-enabled mutations and their testing effectiveness), Sec. 7.3 (RQ3: reliability and improvement of validation metrics), and Sec. 7.4 (RQ4: ablation study of pipeline components).

Table 3. Results of human evaluations. We mark results within $[0, 1)$, $(2, 4]$, and $(4, 5]$ in red, blue, and green, respectively. Scores between 1 and 2 represent poor but not entirely failed alignment, and are left unmarked.

Metric		Contrast	Rotation	VanGogh	Snow_G	Snow_D	Eyes	Library	Clothes	Legs	Glasses
Alignment	LLaVA	0.8	0.0	3.2	2.3	4.3	1.3	1.3	1.9	1.8	3.8
	CogVLM	0.6	0.1	3.1	2.0	3.3	0.7	1.1	1.4	1.3	2.4
	GPT-4V	0.7	0.1	3.1	2.1	3.8	1.8	1.3	2.3	2.1	3.2
	InstP2P	1.2	0.0	2.5	0.6	0.1	0.1	2.6	0.7	1.1	4.2
	Traditional	4.9	4.9	1.7	0.1	4.2	4.1	N/A	N/A	N/A	N/A
Faithfulness	LLaVA	3.0	3.6	3.3	2.5	2.0	3.3	2.9	3.0	3.0	3.5
	CogVLM	2.7	3.5	3.0	2.3	2.0	3.0	2.8	2.5	3.1	3.3
	GPT-4V	3.0	3.7	3.3	2.7	2.1	3.0	3.0	2.8	3.1	3.5
	InstP2P	4.9	1.0	4.2	4.6	5.0	3.3	3.0	4.2	4.5	4.3
	Traditional	5.0	4.9	4.6	4.5	0.2	4.4	N/A	N/A	N/A	N/A
Validity	LLaVA	4.0	4.5	4.1	4.0	2.8	3.1	3.1	3.3	3.6	4.4
	CogVLM	3.6	4.3	3.7	3.8	2.7	2.5	2.9	3.2	3.6	4.0
	GPT-4V	4.0	4.2	3.9	3.8	2.9	2.5	3.1	3.3	3.8	4.5
	InstP2P	4.9	0.9	4.1	4.8	4.9	3.6	2.6	4.2	4.5	4.2
	Traditional	5.0	4.9	4.4	4.5	2.5	4.4	N/A	N/A	N/A	N/A

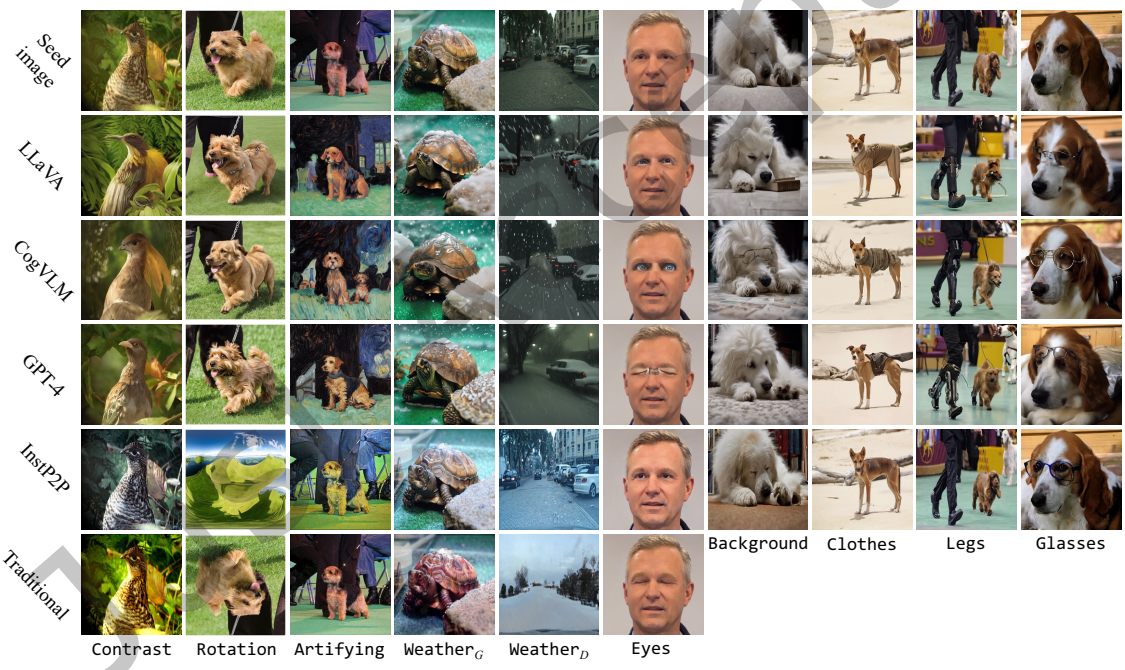


Fig. 4. Mutation examples achieved via traditional method and different MLLMs.

7.1 RQ1: Capabilities in Unifying Traditional Mutations

Table 3 reports the average score over all 200×2 or 250×2 samples for each mutation. Since a zero score indicates that a mutated image fails to satisfy one requirement, we view scores within $[0, 1)$ as failing and mark them in red. We treat scores larger than two as satisfying the requirement. Specifically, scores within $(2, 4]$ are marked in blue, whereas scores within $(4, 5]$ are marked in green to highlight the best method for one mutation.

Result Validation. As mentioned in Sec. 6, we implement a sanity check to rule out answers from participants who likely misunderstood the questions. We report that all participants in our evaluation pass the sanity check. Further, since our final scores are averaged over two participants per mutation, to check whether participants reached consensus, we also measure the inter-annotator agreement using Kappa statistics. We employ the Quadratic Weighted Cohen's Kappa [9] given that in our evaluation, participants' choices are ordinal (i.e., the scores are from 0 to 5, with meaningful distances and orders). The Kappa score ranges from -1 (complete disagreement) to 1 (perfect agreement), and our resulting score is 0.73, indicating a substantial level of agreement between annotators.

Fig. 4 shows examples of different mutations implemented using different methods (more examples are provided in the artifact [5]). Below, we interpret our human study results with these examples.

`Contrast` and `Rotation`. Our results show that MLLMs are unable to implement pixel- and geometrical-level mutations, as indicated by the red scores under the alignment evaluation. Note that pixel- and geometrical-level mutations directly operate on image pixels, whereas the remaining mutations focus on image features (i.e., image representations obtained using neural networks). MLLMs also primarily accept image features, leaving lower-level image semantics untouched. This explains why MLLMs fail in implementing pixel- and geometrical-level mutations, as images before and after these mutations often do not lead to notable changes in image features. Recent MLLM papers have similar observations: MLLMs perform worse in the spatial reasoning of image contents [8, 15, 60].

Different from others, InstP2P fails in the validity evaluation of `Rotation` and the faithfulness evaluation also has an average score of one. As shown in Fig. 4, InstP2P generates a meaningless image when implementing `Rotation`, and we find that similar failures occur in all evaluated samples. Recall as noted in Sec. 3.4, InstP2P differs from other MLLMs by its unidirectional pipeline: to complement the missing details in the text instruction, InstP2P fine-tunes the image generator (i.e., the Diffusion model) with mutation samples. We therefore suspect that the fine-tuning harms the generative capability of the Diffusion model. Hence, when implementing geometrical-level mutations that require spatial reasoning, InstP2P fails to generate valid images. In contrast, as other MLLMs do not modify the image generator, they mostly "transmit" the seed images if they cannot implement a mutation.

`VanGogh`. MLLMs outperform traditional methods in transferring the artistic style of images. Past methods often extract the artistic style from one or a few target images, whose understanding of the target style may be limited. Moreover, traditional style transfer may fail if contents in the seed image and the target have distinct geometrical structures [91]. These two limitations explain why the alignment score of the traditional method is unsatisfactory (i.e., only 1.7). MLLMs encode general knowledge and can better comprehend the target style from their immense amounts of training data.

However, MLLM's faithfulness results are relatively worse than those of traditional methods. As shown in Fig. 4, besides transferring the style, MLLMs also edit other irrelevant semantics in the mutated images. This observation is consistent with our findings mentioned above: MLLMs often focus on high-level features and neglect low-level details. As a result, such details are lost and consequently "inpainted" with random ones when generating mutated images.

`Snow`$_G$ and `Snow`$_D$. InstP2P fails to transfer weather conditions for general images and driving scenes. We attribute the reason to the lack of fine-tuning data. To support transferring weather conditions, the fine-tuning in InstP2P requires mutation variants of the seed image that contain different weather conditions, and such data tuples

are hard to prepare on a large scale. Thus, the fine-tuned Diffusion model in InstP2P is unable to capture the weather conditions, transmitting the input image to the output in most cases, as indicated by the high faithfulness and validity score. The traditional method leverages the cycle-consistency to achieve weather transfer without requiring paired images as the guidance. This paradigm successfully transfers the weather conditions for driving scenes, resulting in a high alignment score. However, it fails to retain the faithfulness (i.e., a 0.2 score), such that mutated images do not preserve irrelevant semantics, as presented in Fig. 4. In addition, the cycle-consistency relies on the similar geometrical structure between the seed and target images; this explains why traditional method is inapplicable to mutate weather conditions in general images whose geometrical structures are largely distinct from driving scenes.

The other three MLLMs, especially GPT-4V, have satisfactory alignment, faithfulness, and validity when transferring weather conditions for both general images and driving scenes. We attribute these to the general knowledge encoded in MLLMs. Although fine-grained details are also modified in MLLM's generated images (see Fig. 4), we envision MLLM's promising application in testing VDL systems where weather conditions are crucial (e.g., auto-driving).

Eyes. Traditional methods outperform MLLMs when implementing perceptual-level mutations. All MLLMs are unable to deliver satisfactory alignment scores. As in Fig. 4, all MLLMs do not close the eyes but replace them with new eyes in most cases. That is, these MLLMs understand that the mutation modifies eyes but cannot precisely mutate them. Moreover, recall that as mentioned in Sec. 4, the traditional implementation of `Eyes` requires specifying the localization of eyes. To make the comparison fair, we also provide the eye locations to MLLMs. However, as shown in Fig. 4, MLLMs fail to generate smoothly mutated images and the edited eyes look unnatural, leading to relatively lower validity scores.

Beyond individual mutations, alignment is consistently low across all types. To understand why, we randomly sampled 100 mutated images with alignment scores below 1 (25 for each of the four MLLM-based mutation methods) for manual inspection. Two researchers independently analyzed each sample to identify misalignment causes, and resolved disagreements through discussion. Generators capture the mutation intent but execute it as *generation* rather than *transformation*. Instead of modifying existing content, they synthesize new content that loosely matches the instruction (e.g., replacing eyes instead of closing them, producing a new scene instead of rotating the original, or even omitting the original subject altogether; see Fig. 4). We attribute this primarily to information loss in the text-to-image encoding process: the generator receives a compressed representation that lacks the original image's fine-grained details, preventing it from performing precise edits [35, 36]. Limited spatial reasoning of current generators [8, 15, 60] may further compound the problem. Importantly, low alignment does not negate testing utility; rather, it sharpens the complementary positioning of traditional and MLLM-based mutations. Traditional methods remain the reliable choice for precise, well-defined mutations (pixel-level, geometric), while MLLMs uniquely enable semantic-replacement mutations that introduce new semantics traditional methods cannot produce at all. The semantic perturbations from MLLMs reach regions of the model's decision space that pixel-level or geometric transformations cannot. Our unique-fault analysis in Sec. 7.2 empirically confirms this separation (Fig. 5), and we translate these observations into actionable guidance in Sec. 8. We discuss directions to improve alignment in Sec. 9.

Answer to RQ1: MLLMs cannot implement pixel- and geometrical-level mutations well, potentially due to the lost information of such low-level semantics when MLLMs extract image features. MLLMs are also unable to generalize perceptual-level mutations. However, MLLMs optimized with the bidirectional pipeline perform better in extending style-level mutations to more general scenarios. They also exhibit better resilience: since they do not modify the image generator, they will not generate invalid images if they cannot achieve a mutation.

7.2 RQ2: Benefits MLLMs Bring to Input Mutations

7.2.1 New Mutations Enabled by MLLMs. As in Table 3, we find that only InstP2P can implement `Library` mutation to a satisfactory level, whereas the other three MLLMs tend not to edit the seed image in `Library` mutation, as indicated by the relatively high faithfulness and validity score (see examples in Fig. 4). However, the trend is on the opposite when implementing `Clothes` and `Legs` mutations, where InstP2P has lower alignment scores but much higher faithfulness and validity scores. For `Glasses` mutation, all MLLMs have satisfactory alignment, faithfulness, and validity scores, and InstP2P outperforms the other three MLLMs.

Overall, these results demonstrate MLLM's unique capability to perform semantic-replacement mutations, which was previously impossible with traditional methods. This capability significantly expands the potential scope of VDL testing by introducing a new dimension of mutations. Notably, certain MLLM pipelines work well for specific mutations. These varied results across different pipelines (see Sec. 3.4) reveal their distinct strengths and limitations. Both optimized pipelines support subtle mutations to subjects i.e., all four MLLMs achieve satisfactory results for `Subject (Minor)` mutations (e.g., `Glasses` in Table 3). However, their capabilities vary in the remaining three mutations: the unidirectional pipeline is more suitable for `Background` mutations (`Library` in Table 3), while the bidirectional pipeline performs better for `Subject (Major)` and `Subject (Moderate)` mutations (`Clothes` and `Legs` in Table 3) that substantially modify the subject. These findings highlight the importance of selecting appropriate MLLM pipelines based on the specific mutation goals, which we discuss further in Sec. 9.

7.2.2 Testing Effectiveness of MLLM-Enabled Mutations. To further study whether MLLM-enabled mutations are useful in VDL testing, we evaluate their fault-revealing capability in this section.

Setup. Our evaluation focuses on *mutation-level* effectiveness, not *implementation-level* performance. That is, we assess whether the new semantic-replacement mutation types enabled by MLLMs can effectively reveal faults in VDL models, rather than comparing different implementation strategies for the same mutation. Our baselines are the fault-revealing rates of traditional mutation types that are applicable to general images. Specifically, we compare the four types of MLLM-enabled, semantic-replacement mutations against the four types of traditional, generalizable mutations: `Pixel`, `Geometric`, `Style (Artistic)`, and `Style (Weather)`. All mutations belonging to these types are evaluated in this section. We exclude `Perceptual` because it is currently limited to face images. Results are reported at the type level (aggregating mutations per Table 2). `Pixel` and `Geometric` mutations are implemented with their original methods (since MLLMs are unable to implement them as shown in Sec. 7.1). `Style (Weather)` and `Style (Artistic)` mutations are implemented using both traditional methods and MLLMs.

VDL-based classification is selected as one representative VDL application. We choose three recent SoTA VDL-based classifiers, Vision Transformer (ViT) [19], SwinTransformer (SwinT) [51], and ConvNeXt [52]. These models are officially released by PyTorch and are trained with ImageNet. Mutated images are fed into a popular tool DeepHunter [77] to detect VDL faults. Since different mutations may change the seed image to different extents, to avoid this, DeepHunter implements a filter to ensure that the number of changed pixels and the modifications of pixel values in all test inputs are at the same level.

Fault-revealing Rates. Results are reported at the mutation-type level by aggregating all mutations within each type (per Table 2). Results of the fault-revealing rates (i.e., the number of VDL faults divided by the number of test inputs) are shown in Table 4. We find that MLLM-enabled mutations achieve fault-revealing rates ranging from 4.3% to 7.5% (the "Average" rows in Table 4), which are comparable to, and sometimes exceeding traditional mutations. The results confirm that MLLM-enabled mutations are a highly effective and complementary addition to the VDL testing toolkit, expanding the mutation space with entirely new fault-revealing capabilities.

Similar to our observations in Sec. 7.1, we also notice performance variations across different MLLMs when they are implementing the same mutation type. We find that these variations are primarily driven by their underlying pipelines. Specifically, the unidirectional pipeline (i.e., InstP2P), which relies on fine-tuning for end-to-end image

Table 4. **Fault-revealing rates, i.e., (#VDL faults)/(#test inputs), of different types of mutations. N/A indicates that a type of mutation is unsupported by MLLMs or traditional methods.**

		Traditional				MLLM-enabled			
Model	**Impl.**	Pixel	Geometric	Style (Artistic)	Style (Weather)	Background	Subject (Major)	Subject (Moderate)	Subject (Minor)
ConvNeXt	LLaVA	N/A*	N/A	9.5%	8.5%	6.0%	8.0%	6.1%	4.0%
	CogVLM			11.5%	8.0%	4.0%	9.2%	8.2%	8.0%
	GPT4V			9.5%	7.0%	6.0%	6.0%	2.1%	4.0%
	InstructP2P			8.5%	5.5%	14.0%	4.2%	1.6%	1.2%
	Traditional	6.0%	0.2%	5.3%	11.0%	N/A	N/A	N/A	N/A
	Average	**6.0%**	**0.2%**	**8.9%**	**8.0%**	**7.5%**	**6.9%**	**4.5%**	**4.3%**
SwinTransformer	LLaVA	N/A	N/A	8.0%	9.0%	4.0%	6.0%	6.1%	4.0%
	CogVLM			14.0%	9.5%	4.0%	10.0%	6.1%	8.0%
	GPT4V			8.0%	9.0%	8.0%	6.0%	6.1%	4.0%
	InstructP2P			12.0%	8.0%	12.0%	1.2%	1.8%	1.1%
	Traditional	7.5%	0.2%	10.7%	11.0%	N/A	N/A	N/A	N/A
	Average	**7.5%**	**0.2%**	**10.5%**	**9.3%**	**7.0%**	**5.8%**	**5.0%**	**4.3%**
VisionTransformer	LLaVA	N/A	N/A	9.5%	9.5%	4.0%	2.0%	6.1%	6.0%
	CogVLM			12.0%	9.0%	4.0%	10.0%	8.2%	8.0%
	GPT4V			11.0%	6.5%	6.0%	6.0%	5.1%	4.0%
	InstructP2P			10.5%	6.5%	14.0%	4.0%	2.4%	3.5%
	Traditional	8.9%	0.2%	7.0%	13.3%	N/A	N/A	N/A	N/A
	Average	**8.9%**	**0.2%**	**10.0%**	**9.0%**	**7.0%**	**5.5%**	**5.5%**	**5.4%**

* N/A indicates that the mutation type is unsupported by the corresponding method (MLLMs or traditional).

transformation, excels at global edits. For instance, it achieves a high fault rate of 14.0% on the Background mutation. In contrast, bidirectional pipelines use MLLMs to first generate detailed text prompts, then feed them to image generators. This design proves more effective for local, subject-specific modifications. As shown in Table 4, bidirectional pipelines outperform InstP2P on mutations like `Subject (Major)` and `Subject (Moderate)`, which require precise subject-focused edits. Interestingly, even though different MLLMs (e.g., GPT-4V vs. CogVLM), when being incorporated into the same bidirectional pipeline, exhibit variations in fault detection effectiveness due to the differences in their comprehension capabilities. The gaps between different MLLMs are less pronounced than those between different pipeline architectures. This finding suggests that pipeline design plays a more crucial role in determining mutation effectiveness than the specific MLLM used.

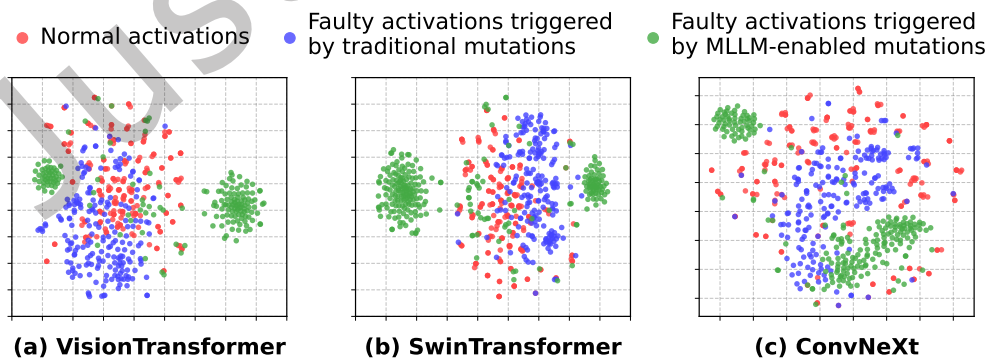


Fig. 5. **Visualization of normal and faulty activations triggered by different mutations.**

Unique Faults. While Table 4 demonstrates that MLLM-enabled mutations are effective at revealing faults, a critical question remains: do these mutations expose genuinely unique faults, or do they merely trigger the same faults as traditional mutations through different visual variations? This distinction is essential for our mutation-level evaluation. If MLLM-enabled mutations trigger the same underlying faults as traditional mutations, they would offer limited value in VDL testing. Conversely, if they expose distinct failure patterns, they represent a genuinely valuable expansion of the testing's capability. To answer this question, we analyze the internal behaviors of VDL models when processing fault-revealing and normal inputs. Following established practices in DNN testing [56], we employ t-SNE to visualize the neural activation patterns extracted from the penultimate layer (whose outputs are leveraged for the final prediction and often reflect the internal logic) of each VDL model. This analysis enables us to examine whether failures induced by different mutation types exhibit distinct internal activations, which would indicate fundamentally different faulty patterns inside the VDL models.

Fig. 5 presents the visualization results in three VDL models' activations spaces, where each point denotes the activation of one input image. Blue and green colors correspond to fault-revealing images generated by traditional and MLLM-enabled mutations, respectively, while red color indicates correctly classified normal images. The visualization reveals a striking pattern: failures caused by MLLM-enabled mutations form tight, concentrated clusters that are spatially separated from both normal activations and failures triggered by traditional mutations. In other words, MLLM-enabled mutations expose unique classes of faulty patterns that traditional mutations cannot. To explain, the new semantics brought by MLLM-enabled mutations, through replacement (e.g., adding glasses, dressing with clothes), could reach decision logics or pathways inaccessible to traditional mutations. This further justifies that MLLM-enabled mutations are a valuable and complementary addition to VDL testing compared to traditional mutations.

Answer to RQ2: MLLMs enable a new class of semantic-replacement mutations that traditional methods cannot implement. These new mutations not only achieve fault-revealing rates comparable to traditional mutations, but also expose unique faults unreachable by prior approaches, significantly expanding the scope and effectiveness of VDL testing. The pipelines of employing MLLMs for mutations also matter: unidirectional pipelines often excel at global edits, while bidirectional pipelines could better handle local subject modifications.

7.3 RQ3: Reliability of Validation Metrics

Aligned to the three aspects considered in our human study, we evaluate whether existing validation metrics can reflect the quality of test inputs generated by MLLM-enabled mutations and capture their differences. This section uses the full evaluation set for MLLM-enabled mutations introduced in Sec. 5.3. Each MLLM-enabled mutation is instantiated on 250 seed images. For each reported mutation type in Table 2, we pool the two constituent mutations in that type, namely the foundational Dog Breed mutation and the ImageNet mutation with class specific instructions, and report one type level result per MLLM.

7.3.1 Validity Metrics. Following existing work [27, 82], we take two popular metrics, Inception Score (IS) [64] and Frechet Inception Distance (FID) [30], to measure the validity of the mutated images. Both IS and FID are data-driven: they use a set of real-world images as the reference and measure the statistical similarity between mutated images and the reference images. A lower FID value indicates a better result, whereas a higher IS value is better. To assess whether the metrics can capture the differences in MLLM's capabilities, we also define Normalized Relative Gain (NRG) in the sense that different metrics may operate on different scales. Given the i-th MLLM's validity score S_i, the worst score among all evaluated MLLMs S_{worst}, and the value range a metric

can reach S_{range}, the NRG for the i-th MLLM is defined as:

$$NRG_i = \frac{|S_i - S_{worst}|}{S_{range}} \tag{2}$$

Both IS and FID have the minimal value of 0. The maximal value of IS is the number of classes in its reference dataset [10], i.e., 1000 for ImageNet. FID does not have a maximal value in theory, but in practice, recent work [44] has reported FID values up to 300, so we set 300 as the maximal value for FID.

Table 5. **Results of existing validity metrics (IS, FID) with NRG in parentheses.**

		Background	Subject (Major)	Subject (Moderate)	Subject (Minor)
IS	LLaVA	11.92 (**+0.0005**)	12.18 (**+0.0004**)	11.50 (**+0.0000**)	12.08 (**+0.0015**)
	CogVLM	11.42 (**+0.0000**)	11.78 (**+0.0000**)	12.02 (**+0.0005**)	12.08 (**+0.0015**)
	GPT-4V	12.12 (**+0.0007**)	11.83 (**+0.0001**)	11.63 (**+0.0001**)	12.32 (**+0.0018**)
	InstP2P	12.40 (**+0.0010**)	13.39 (**+0.0016**)	13.59 (**+0.0021**)	10.57 (**+0.0000**)
FID	LLaVA	90.28 (**+0.0666**)	91.69 (**+0.0036**)	97.38 (**+0.0591**)	95.25 (**+0.1141**)
	CogVLM	95.25 (**+0.0500**)	92.77 (**+0.0000**)	95.62 (**+0.0650**)	95.08 (**+0.1146**)
	GPT-4V	90.17 (**+0.0669**)	92.12 (**+0.0022**)	98.18 (**+0.0564**)	92.26 (**+0.1240**)
	InstP2P	110.25 (**+0.0000**)	84.05 (**+0.0291**)	115.11 (**+0.0000**)	129.47 (**+0.0000**)

As shown in Table 5, both IS and FID appear to be random and unable to reflect the capabilities of different MLLMs in implementing valid mutations. In many cases, the two metrics even produce contradictory results. For example, in `Subject (Moderate)` mutations, InstP2P achieves the best IS score but the worst FID score among all MLLMs. These conflicting signals prevent practitioners from reliably selecting appropriate MLLMs for specific mutations. Moreover, they fail to capture subtle yet meaningful performance gaps among MLLMs: the NRG values are extremely small (ranging from +0.0000 to +0.0021 for IS), making it nearly impossible to differentiate MLLMs' varied abilities across different mutation types. Overall, observations in Table 5 indicate the inaccuracy and unreliability of existing validity metrics when handling MLLM-enabled mutations. The primary reason, as noted in prior work [43, 63], is that metrics like IS and FID conflate two distinct aspects: validity and diversity. This was less of an issue for traditional mutations (e.g., Rotation), which produce highly deterministic and similar outputs. In contrast, semantic-replacement mutations enabled by MLLMs are non-deterministic and generate a wide variety of valid mutants for a single instruction. As a result, the validity metric's results can be falsely improved by diversified mutants in MLLM-enabled mutations.

7.3.2 Improving Validity Metrics. As discussed before, the conflation of validity and diversity in existing metrics makes them inaccurate and unreliable for MLLM-enabled mutations. To address this issue, we recommend using metrics that could separate validity from diversity. In this section, we adopt the `Precision & Recall` metrics that are particularly designed for this purpose [43, 63]. Specifically, `Precision` reflects the validity of images, while `Recall` measures their diversity. Both metrics range from 0 to 1, with higher values indicating better performance, and are naturally normalized to allow easy comparison. `Precision` and `Recall` are computed on two image sets in a learned feature space via k-NN manifold estimation [43]. Concretely, all images are projected into VGG-16 [67] feature space, and for each image the k-nearest neighbors (k=3, following the default in [43]) define a hypersphere whose union approximates the support of a distribution. `Precision` is the fraction of generated images that fall within the reference manifold, and `Recall` is the fraction of reference images that fall within the generated manifold. Following prior VDL testing practice [82], for each mutation, the reference set is its seed images and the generated set is the corresponding mutant images produced by one MLLM; type-level results average over the constituent mutations.

Table 6 reports the Precision (validity) results, which are broadly consistent with our human study findings. In particular, they recover the main cross-type conclusions from the human study: InstP2P is the best in Subject (Major) and Subject (Moderate) but is the worst in Background. The NRG values shown in parentheses are distinguishable enough to reflect the subtle differences among MLLMs' varied capabilities in implementing different mutations. We also report Recall (diversity), which is marked in gray, in Table 6; these high diversity values further justify why the conflation issue makes existing metrics unreliable for MLLM-enabled mutations.

Table 6. **Results using the improved validity metric (Precision). To ease the comparison with and interpretation of Table 5, we report NRG values in parentheses and Recall (diversity) values in gray.**

		Background	Subject (Major)	Subject (Moderate)	Subject (Minor)
LLaVA	Precision	0.36 (**+0.03**)	0.86 (**+0.02**)	0.80 (**+0.03**)	0.86 (**0**)
	Recall	0.70	0.44	0.28	0.32
CogVLM	Precision	0.35 (**+0.02**)	0.87 (**+0.03**)	0.85 (**+0.08**)	0.87 (**+0.01**)
	Recall	0.39	0.28	0.30	0.28
GPT-4V	Precision	0.32 (**0**)	0.84 (**0**)	0.77 (**0**)	0.92 (**+0.06**)
	Recall	0.62	0.34	0.26	0.34
InstP2P	Precision	0.32 (**0**)	0.93 (**+0.09**)	0.88 (**+0.11**)	0.89 (**+0.03**)
	Recall	0.47	0.77	0.81	0.74

7.3.3 Embedding-Based Metrics for Alignment and Faithfulness. As mentioned in Sec. 3.2, for mutations lacking explicit formulas (i.e., style-level, perceptual-level, and MLLM-enabled mutations), existing works propose to measure them via image embeddings. Following recent MLLM papers [29, 55, 66, 87], we obtain image embeddings via the SoTA embedding model CLIP [59]. The CLIP can obtain unified embeddings for image semantics and text descriptions, enabling their similarity measurement. We first prepare the textual description of the anticipated mutation result for each mutation. This is done by leveraging an LLM to generate textual descriptions for the expected mutation results based on the mutation instructions. For example, when mutating a dog image with Glasses, the mutated image should have "a dog with glasses". We then use the CLIP to extract the embedding vectors for the mutated image and its corresponding textual description; their cosine similarity is adopted to quantify the alignment of the mutation.

The faithfulness evaluation requires separating the local and fine-grained semantics such as eyes and legs from the image, which is challenging especially when the locations and postures of the dog are diverse. Therefore, we design a coarse-grained metric that only considers an image's background and subject. Specifically, following [34], we employ the SoTA subject extractor InSPyReNet [41] to separate the subjects and background for the seed and mutated images. For Library and Background_G, we use the similarity of the two subjects in the seed and mutated image as the faithfulness result, whereas for ⑲-㉔, we use the similarity of two backgrounds for faithfulness. Since semantic-replacement mutations may alter the subject's outline (e.g., adding clothes may change the dog's shape), we compute the similarity as the cosine similarity between CLIP image embeddings of the extracted subjects/backgrounds.

Table 7. **Results of the existing alignment metric. NRG is omitted since the metric itself is normalized.**

	Background	Subject (Major)	Subject (Moderate)	Subject (Minor)
LLaVA	0.51	0.41	0.37	0.41
CogVLM	0.51	0.43	0.37	0.41
GPT-4V	0.51	0.41	0.35	0.41
InstP2P	0.52	0.46	0.39	0.43

Table 8. **Results of the existing faithfulness metric. NRG is omitted since the metric itself is normalized.**

	Background	Subject (Major)	Subject (Moderate)	Subject (Minor)
LLaVA	0.88	0.85	0.85	0.86
CogVLM	0.87	0.85	0.85	0.86
GPT-4V	0.88	0.85	0.85	0.85
InstP2P	0.85	0.90	0.88	0.87

Table 7 and Table 8 present the results of alignment and faithfulness evaluations for four MLLM-enabled mutations using popular embedding-based metrics, respectively. The alignment results of `Background` are relatively better than that of `Subject (Major)`, `Subject (Moderate)`, and `Subject (Minor)`, indicating that the metric can detect the mutated background, but is unable to detect the mutated finer-grained semantics on subjects. The high faithfulness results show that the embedding-based metric can reflect the (ought-to-be) unchanged semantics. Nevertheless, unlike our human studies that show notable differences in alignment and faithfulness results of different MLLMs, the two numerical metrics cannot distinguish their varied results, despite that similar methods show promising results for traditional mutations [55].

Note that traditional mutations like style- and perceptual-level mutations only apply to specific domains, and accordingly, their corresponding image embeddings can be limited to the target domain to improve precision. E.g., when measuring mutations applied to face photos, existing works leverage face generators to obtain face embeddings and do not consider non-face images. Similarly, when measuring the achieved style-level mutations, prior works leverage style extractor to specifically compute the style distance. However, MLLM-enabled mutations are applicable to more general images. Although CLIP can obtain embeddings for diverse images from different domains, these embeddings consequently sacrifice their precision; similar observations are also noted in recent research [79, 88].

7.3.4 Improving Alignment and Faithfulness Metrics. The limited precision of CLIP embeddings on general images points to a clear direction: we need finer-grained metrics that can capture subtle semantic differences. Below, we introduce our improved faithfulness and alignment metrics.

Improving Faithfulness Metrics. The core limitation of existing faithfulness metrics lies in their reliance on global image embeddings. Such coarse-grained representations inevitably miss local semantic changes. For example, when a mutation modifies a subject's glasses, global embeddings average this change across the entire image, diluting its impact. To capture such fine-grained changes, we adopt a patch-based approach. We partition each image into 4×4 patches and extract CLIP embeddings for each patch independently. The faithfulness score is then computed as the average cosine similarity across all corresponding patch pairs between the original and mutated images. This design enables the metric to detect local semantic changes that global embeddings overlook.

Table 9 shows that our patch-based metric effectively distinguishes MLLMs by their faithfulness. InstP2P achieves substantially higher scores on `Subject (Major)` (0.84), `Subject (Moderate)` (0.82), and `Subject (Minor)` (0.83) mutations, demonstrating superior sensitivity to fine-grained visual changes. This is consistent with our findings in Table 3, where human evaluators also rated InstP2P higher for faithfulness. In contrast, other MLLMs (LLaVA, CogVLM, and GPT-4V) exhibit consistent but moderate performance across all mutation types (0.67–0.72), suggesting they capture these changes less precisely, which aligns with our human study results in Table 3.

Improving Alignment Metrics. While patch-based analysis addresses faithfulness measurement, alignment metrics face a different challenge: visual embeddings struggle to capture the semantic alignment between mutation instructions and the actual changes. For instance, distinguishing whether a mutated image shows "winter clothes" versus "summer clothes" requires understanding textual categories (i.e., "winter" vs. "summer") rather than visual similarity alone. We address this challenge by shifting from visual comparison to textual analysis. Our key insight

Table 9. **Results using the improved faithfulness metric. NRG is omitted since the metric itself is normalized.**

	Background	Subject (Major)	Subject (Moderate)	Subject (Minor)
LLaVA	0.71	0.71	0.67	0.70
CogVLM	0.68	0.70	0.68	0.69
GPT-4V	0.72	0.71	0.68	0.69
InstP2P	0.72	0.84	0.82	0.83

is that textual descriptions can express semantic attributes more precisely than visual embeddings can capture them.

Specifically, we leverage LLaVA-NeXT [46] to generate textual descriptions of mutated images, and then employ spaCy [32] to extract semantic attributes (nouns and adjectives via part-of-speech tagging) from these descriptions. For each mutation, we predefine a set of target keywords that characterize a successful mutation (e.g., {"library", "bookshelf", "books"} for the `Library` mutation). The alignment score is then computed as the fraction of target keywords that appear among the extracted attributes of the generated description. This approach requires one additional safeguard. Some MLLMs may generate images that completely replace the original subject. For example, given the instruction "change the background to a library" for a dog image, an MLLM might generate a pure library scene without the dog. To rule out such failures, we first verify that the generated description contains the original subject. Only images passing this check proceed to alignment evaluation; others receive a score of zero.

Table 10 demonstrates that our text-based metric successfully differentiates MLLM performance. In particular, all MLLMs achieve significantly higher alignment scores on `Glasses` (0.65–0.87) compared to other mutations (0.11–0.54), suggesting that minor semantic subject mutations are inherently easier than modifying backgrounds or body parts; this is also consistent with our human judgments in Table 3. Second, InstP2P outperforms the other three MLLMs on `Library` (0.24 vs. 0.11–0.13), indicating its superior ability to generate context-aware backgrounds. These results are the same as our human study findings in Sec. 7, confirming the metric's effectiveness.

Table 10. **Results using the improved alignment metric. NRG is omitted since the metric itself is normalized.**

	Background	Subject (Major)	Subject (Moderate)	Subject (Minor)
LLaVA	0.11	0.40	0.27	0.83
CogVLM	0.13	0.45	0.11	0.65
GPT-4V	0.13	0.54	0.24	0.77
InstP2P	0.24	0.21	0.15	0.87

Answer to RQ3: Prior validation metrics fall short when applied to MLLM-enabled mutations, but we identify clear paths for improvement. First, validity metrics like IS and FID conflate validity with diversity. This conflation misleads evaluation especially for MLLM-enabled mutations which non-deterministically generate diverse outputs. We therefore recommend using the `Precision & Recall` metrics [43, 63] that separate validity and diversity. Our evaluation shows that `Precision` scores recover the main trends of human judgments and capture MLLMs' subtle yet varied capabilities. Second, existing embedding-based metrics using global image features fail to capture fine-grained semantic changes. We demonstrate that two complementary improvements address this limitation: (1) patch-based CLIP embeddings effectively distinguish faithfulness across MLLMs by detecting local changes, and (2) the text-based extraction of semantic attributes successfully measures alignment by comparing mutation requirements against generated image descriptions. Both improved metrics show consistent results with human judgments and effectively differentiate MLLM performance.

7.4 RQ4: Ablation Study: Different Pipelines

In RQ2, we observe that pipeline architectures differ markedly in mutation quality. However, each pipeline is a multi-component system: the bidirectional pipeline uses enriched MLLM prompts (Sec. 3.4), while the unidirectional pipeline uses a fine-tuned image generator. Observing only end-to-end performance cannot tell us whether differences stem from prompt design or generator capability. A component-level analysis can identify which factor affects mutation quality the most, directly informing engineering decisions: whether to invest effort in prompt refinement or generator fine-tuning, and which prompting MLLM to use when enriched prompting is enabled, given substantial cost differences (e.g., proprietary GPT-4V vs. open-source LLaVA). We therefore conduct a structured ablation study to isolate each component's contribution.

We vary two design factors in a 2×2 ablation: prompt type (simple vs. enriched) and generator type (pre-trained vs. fine-tuned). This yields four configurations: (1) Baseline (pre-trained + simple), (2) Bidirectional (pre-trained + enriched), (3) Unidirectional (fine-tuned + simple), and (4) Combined (fine-tuned + enriched). Configurations (2) and (4) use an MLLM for prompt enrichment; we test three prompting MLLMs (LLaVA, CogVLM, GPT-4V) in each. This design isolates the effect of each optimization as well as their interaction. We reuse the full dataset from RQ2 and RQ3. We evaluate using the metrics from RQ3: `Precision` for validity, patch-based similarity for faithfulness, and text-based similarity for alignment. Results appear in Table 11.

We compare ablation settings in Table 11. Prompt enrichment and generator fine-tuning each independently improve quality along different dimensions: enrichment yields the largest gain in alignment (0.19→0.39), while fine-tuning yields the largest gain in faithfulness (0.71→0.81) by learning to preserve unchanged regions; both raise validity from 0.58 to 0.79. Separately, within a fixed pipeline, the choice of prompting MLLM has limited impact (validity varies by $\leq$0.05), whereas switching pipeline configuration changes validity by 0.31–0.33, confirming that pipeline architecture is the dominant factor. However, combining both optimizations proves counterproductive: validity drops to 0.48 and alignment to 0.09, below either optimization alone. We note that the fine-tuned generator in InstP2P was trained on concise, single-sentence instructions [11], whereas enriched prompts are substantially longer and multi-clause, creating a *prompt-distribution mismatch* between the enriched input and the generator's training distribution. Similar observations have been documented in related work on fine-tuned diffusion models [62]. Given this constraint, practitioners should use each optimization separately, choosing the pipeline based on mutation scope and preferring open-source MLLMs to reduce cost, rather than combining them. We elaborate on these guidelines in Sec. 8. Resolving the mismatch itself remains a promising direction for future work. Potential strategies include broadening the generator's training distribution with diverse instruction formats [85], summarizing enriched prompts into concise directives before generation, or adopting adapter-based

Table 11. **Ablation study comparing the effects of prompt enrichment, fine-tuning, and prompt-MLLM choice on mutation quality.**

Configuration	Generator†	Prompt MLLM	Validity	Faithfulness	Alignment
Baseline	Pre-trained + Simple	—	0.58	0.71	0.19
Bidirectional	Pre-trained + Enriched	LLaVA	0.81	0.69	0.41
		CogVLM	0.76	0.69	0.34
		GPT-4V	0.81	0.70	0.42
		Avg.	0.79	0.69	0.39
Unidirectional	Fine-tuned + Simple	—	0.79	0.81	0.37
Combined	Fine-tuned + Enriched	LLaVA	0.46	0.75	0.10
		CogVLM	0.48	0.77	0.08
		GPT-4V	0.49	0.76	0.10
		Avg.	0.48	0.76	0.09

† Fine-tuned generator refers to InstP2P. Pre-trained image generator refers to Standard Stable Diffusion.

fine-tuning methods that are more robust to input variation [33]. Evaluating these strategies in the context of VDL mutation testing warrants further investigation.

Answer to RQ4: Prompt enrichment and generator fine-tuning each improve mutation quality when applied separately, but combining enriched prompts with a generator fine-tuned on simple prompts produces a negative interaction that lowers validity and alignment, consistent with a prompt–training mismatch. Among the factors we evaluate, prompt enrichment yields the largest gain in alignment, fine-tuning yields the largest gain in faithfulness, and prompting-MLLM choice has only a minor effect ($\leq$0.05).

8 Practical Insights and Implications

Based on our evaluation, we distill five actionable insights for SE developers and researchers, each targeting a concrete engineering decision in deploying MLLM-based mutations.

Insight 1: Combine Traditional and MLLM Mutations to Cover Non-Overlapping Fault Classes. Traditional methods remain the better choice for pixel-level, geometric, and perceptual mutations, where determinism and controllability are essential and MLLMs lack the spatial precision for faithful edits (Sec. 7.1). MLLM mutations, in turn, enable semantic-replacement mutations (e.g., adding accessories, replacing backgrounds) and semantics-transfer mutations (e.g., style and weather changes) that are infeasible with traditional techniques. Both families achieve comparable fault-revealing rates (4.3%–7.5% for MLLM-enabled vs. 0.2%–10.5% for traditional; Table 4), yet the faults they expose are non-overlapping: MLLM-triggered faults form spatially distinct clusters in VDL activation space (Fig. 5). Neither family alone suffices for comprehensive testing.

Insight 2: Use Open-Source MLLMs for Prompt Enrichment in Bidirectional Pipelines. In bidirectional pipelines, an MLLM first enriches the brief mutation instruction into a detailed prompt before the diffusion generator executes the edit (Sec. 7.4). For this prompting role, practitioners need not invest in proprietary APIs. Within a fixed pipeline, the choice of prompting MLLM has negligible impact: switching MLLMs varies validity by $\leq$0.05 and faithfulness by $\leq$0.01 (Table 11), whereas switching the pipeline configuration changes validity by 0.31–0.33. In bidirectional configurations, the open-source LLaVA matches the proprietary GPT-4V in both validity (0.81) and alignment (0.41 vs. 0.42). CogVLM scores slightly lower (validity 0.76, alignment 0.34) but remains a viable alternative.

Insight 3: Use Unidirectional Pipelines for Global Mutations and Bidirectional for Subject-Level Mutations. Use the unidirectional pipeline (fine-tuned generator, simple prompts) for global, scene-level mutations such as background replacement, where it achieves the highest fault-revealing rates (Table 4). Use the bidirectional pipeline (pre-trained generator, enriched prompts) for subject-focused mutations and for weather and style transfer, where it produces more faithful results (Sec. 7.2). Crucially, do not combine fine-tuned generators with enriched prompts: the two optimizations each improve quality independently, but their combination drops validity from 0.79 to 0.48 and alignment from 0.39 to 0.09 (Table 11), because enriched prompts fall outside the concise-instruction distribution on which the fine-tuned generator was trained [11] (Sec. 7.4).

Insight 4: Replace IS and FID with Precision, Patch-CLIP, and Text Matching for MLLM Mutations. Conventional generative metrics (IS, FID) evaluate distributional quality, but mutation testing requires assessing conditional editing quality: whether the edit matches the instruction while preserving the rest of the image. Because MLLM mutations are non-deterministic, IS and FID conflate editing quality with output diversity, producing contradictory rankings across MLLMs (Sec. 7.3). Global CLIP cosine similarity also fails to capture fine-grained differences (Table 7, Table 8). Instead, researchers should use dimension-specific metrics: `Precision` for validity, patch-based CLIP similarity for faithfulness, and text-based keyword matching for alignment, as validated in Sec. 7.3.

Insight 5: Filter Subject-Major and Subject-Moderate Mutants on Alignment Before Downstream Testing. Minor-subject mutants are reliable without additional filtering, but subject-major and subject-moderate mutants require explicit alignment checks. Our human evaluation reveals a validity-alignment gap of 1.9–2.3 points (on a 5-point scale) for these two categories, versus only 0.9 for minor-subject mutations (Table 3), meaning mutants that pass validity checks can still fail alignment. For instance, MLLMs may replace a subject's eyes rather than closing them, or modify background details while transferring a style. Feeding such unfiltered mutants into downstream testing confounds fault attribution, as detected failures may stem from generation artifacts rather than the intended perturbations (Sec. 7.1). Practitioners should validate these mutants using the dimension-specific metrics from Insight 4 and apply stricter alignment thresholds.

9 Discussion and Threats to Validity

Our results clarify both the promise and the current limitations of MLLM-based mutations in VDL testing. While these pipelines already support some testing scenarios in an out-of-the-box manner, substantial challenges remain, especially in alignment and controllability. In this section, we first reflect on lessons learned from conducting this study, then discuss technical directions for narrowing the alignment gap, describe our reusable benchmark for future research, and address threats to validity.

Lessons Learned. Beyond the practical guidance in Sec. 8, this study yielded three broader observations for future SE research on MLLM-based testing. First, fine-tuning data acts as an implicit contract on the input format. The prompt-distribution mismatch in Sec. 7.4 generalizes: whenever a fine-tuned model receives inputs from another component, practitioners must verify that outputs conform to the model's training distribution, even if both components work well independently. This lesson extends beyond VDL testing to any SE pipeline that composes independently developed ML components; integration testing at component boundaries is essential even when unit-level performance is satisfactory.

Second, the diversity and non-determinism of MLLM mutations make automated evaluation systematically harder than for traditional mutations (Sec. 7.3), even with improved metrics. The same mutation instruction can yield semantically different outputs across runs, which challenges both reproducibility and oracle design. Future studies of MLLM-generated test inputs should include targeted human validation on a representative subset and report variance across repeated generations. This also implies that directly comparing two MLLM-based mutation tools on a single run may be misleading without accounting for output variance.

Third, the validity-alignment gap (Sec. 7.1) reflects a fundamental architectural limitation of current diffusion models, not an engineering misconfiguration. This was not resolved by any pipeline variant or prompting strategy we tested, indicating that prompt engineering and pipeline optimization alone cannot close this gap. Progress requires advances in the underlying generative architectures, such as spatially grounded editing (discussed below). For the SE community, this means that benchmarking new generators on alignment fidelity, not just visual quality, is critical for advancing MLLM-based testing.

Narrowing the Alignment Gap. Our findings in Sec. 7.1 suggest that current pipelines still execute many edit instructions as *generation* tasks rather than *transformation* tasks, primarily due to information loss in text-to-image encoding and limited spatial reasoning. Although developed for general image editing, several recent techniques offer complementary building blocks to narrow this gap. MLLM-guided editors such as MGIE [24] and SmartEdit [36] improve instruction interpretation, reducing semantic drift caused by lossy encoding. Diffusion-control methods like Prompt-to-Prompt [29] manipulate cross-attention to alter targeted concepts while preserving scene structure, helping mitigate information loss during conditioning. Dataset resources such as MagicBrush [85] provide paired edit-instruction supervision that encourages transformation-faithful outputs. Explicit spatial grounding via segmentation models such as SAM [42] can localize edits and compensate for limited spatial reasoning. Adapting these techniques to VDL mutation testing remains an open challenge, as mutations require controlled, auditable transformations where the scope of change is precisely bounded.

Reusable Benchmark. Our work establishes a reusable benchmark to accelerate SE research on AI testing. The rapid evolution of MLLMs has outpaced the development of standardized evaluation frameworks, making it difficult to compare new approaches or measure progress; our benchmark directly fills this gap. We release a comprehensive artifact with three key components. First, our dataset contains over 28,000 generated mutants from the 24 mutations in our full set, each instantiated on 250 seed images under its applicable implementations. Second, we provide complete human evaluation data with 2,300 annotated images, each scored for validity, faithfulness, and alignment by Ph.D. students experienced in testing and VDL systems. Third, we include full implementations of both unidirectional and bidirectional pipelines and our proposed validation metrics. Together, these components enable principled extension of our work. Researchers can assess new MLLMs and new mutation techniques against our baselines using the same evaluation protocol, validate new metrics against our human annotations without additional manual studies, and conduct MLLM-driven VDL testing at scale using our ready-to-use implementations.

Threats to Validity. First, the human study may be biased, and to mitigate this threat, we have involved 20 participants with relevant expertise, and each question is evaluated by different participants to reduce personal bias. We implement blinded sanity checks to rule out answers from participants who likely misunderstand the questions. Results of Kappa statistics also indicate that our participants reach substantial agreement. We believe that these are sufficient to guarantee the credibility of our study.

Second, our findings may not generalize to other pipelines or MLLMs. Both available pipelines are considered in our evaluations, and we incorporate four SoTA MLLMs into them to implement 24 mutations that cover all categories in existing VDL testing literature. These evaluated cases are comprehensive enough to alleviate potential bias due to the selection of mutation types or MLLMs.

Third, the reliability of existing validation metrics for MLLM-based mutations is also evaluated. To make the selected metrics representative, we consider metrics that are widely adopted and have been proven effective in existing testing work. We believe this setup is sufficient to reflect the common issues in validation metrics tailored for traditional mutations. We also open-source our artifact for use and extension [5].

10 Conclusion

This paper presents an in-depth study on MLLM-based mutations in VDL testing and reveals the complementary positioning of MLLMs and traditional mutations. We provide insights into the (in-)adequacy of MLLMs for VDL testing, the specialties of different optimizations and mutation pipelines for MLLMs, and the limitations of existing validation metrics when handling MLLM-based mutations. Accordingly, we propose new automated metrics that show trends consistent with human judgments. We also leverage our dataset, baselines, and evaluation framework to establish a reusable benchmark for future research. Our findings provide actionable guidance for practitioners to design and deploy MLLM-based mutation pipelines in different testing scenarios.

Acknowledgments

We sincerely thank the editors and the anonymous reviewers for their valuable feedback and guidance. This paper was supported in part by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (No. 16214723 and No. C6015-23G) and an ITF grant under the contract TS/161/24FP. We thank HKUST Fok Ying Tung Research Institute and National Supercomputing Center in Guangzhou Nansha Sub-center for computational resources.

References

[1] 2018. Kaggle Competition: Dog Breed Identification. https://www.kaggle.com/competitions/dog-breed-identification.
[2] 2019. CodeQL. https://codeql.github.com/.
[3] 2023. DALLE. https://openai.com/dall-e-2.
[4] Accessed: 2024. Amazon Mechanical Turk. https://www.mturk.com/.
[5] Created: 2025. Research Artifact. https://sites.google.com/view/llm-testing-benchmark.
[6] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
[7] Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program Synthesis with Large Language Models. (2021). arXiv:2108.07732
[8] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi (Eds.). Association for Computational Linguistics, Nusa Dua, Bali, 675–718. doi:10.18653/v1/2023.ijcnlp-main.45
[9] Arie Ben-David. 2008. Comparison of classification accuracy using Cohen's Weighted Kappa. *Expert Systems with Applications* 34, 2 (2008), 825–832.
[10] Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. 2021. Evaluation metrics for conditional image generation. *International Journal of Computer Vision* 129, 5 (2021), 1712–1731.
[11] Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18392–18402.
[12] Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The (R) Evolution of Multimodal Large Language Models: A Survey. *arXiv preprint arXiv:2402.12451* (2024).
[13] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. 2023. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–10.
[14] Wei-Ge Chen, Irina Spiridonova, Jianwei Yang, Jianfeng Gao, and Chunyuan Li. 2023. Llava-interactive: An all-in-one demo for image chat, segmentation, generation and editing. *arXiv preprint arXiv:2311.00571* (2023).
[15] Anthony G Cohn and Jose Hernandez-Orallo. 2023. Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of LLMs. *arXiv preprint arXiv:2304.11164* (2023).
[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.
[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 248–255.

[18] Yinlin Deng, Chunqiu Steven Xia, Haoran Peng, Chenyuan Yang, and Lingming Zhang. 2023. Large language models are zero-shot fuzzers: Fuzzing deep-learning libraries via large language models. In *Proceedings of the 32nd ACM SIGSOFT international symposium on software testing and analysis*. 423–435.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[20] Chengbin Du, Yanxi Li, Zhongwei Qiu, and Chang Xu. 2024. Stable diffusion is unstable. *Advances in Neural Information Processing Systems* 36 (2024).

[21] Isaac Dunn, Hadrien Pouget, Daniel Kroening, and Tom Melham. 2021. Exposing previously undetectable faults in deep neural networks. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 56–66.

[22] Zhiyu Fan, Xiang Gao, Martin Mirchev, Abhik Roychoudhury, and Shin Hwei Tan. 2023. Automated repair of programs from large language models. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 1469–1481.

[23] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2023. Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=PUIqjT4rzq7

[24] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. 2024. Guiding Instruction-based Image Editing via Multimodal Large Language Models. In *International Conference on Learning Representations (ICLR)*.

[25] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

[26] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

[27] Fabrice Harel-Canada, Lingxiao Wang, Muhammad Ali Gulzar, Quanquan Gu, and Miryung Kim. 2020. Is neuron coverage a meaningful measure for testing deep neural networks?. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 851–862.

[28] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems* 33 (2020), 9841–9850.

[29] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626* (2022).

[30] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*. 6626–6637.

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[32] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). doi:10.5281/zenodo.1212303

[33] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=nZeVKeeFYf9

[34] Miao Hua, Jiawei Liu, Fei Ding, Wei Liu, Jie Wu, and Qian He. 2023. DreamTuner: Single Image is Enough for Subject-Driven Generation. *arXiv preprint arXiv:2312.13691* (2023).

[35] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. 2024. Diffusion Model-Based Image Editing: A Survey. *arXiv preprint arXiv:2402.17525* (2024).

[36] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. 2023. SmartEdit: Exploring Complex Instruction-based Image Editing with Multimodal Large Language Models. *arXiv preprint arXiv:2312.06739* (2023).

[37] Naman Jain, Skanda Vaidyanath, Arun Shankar Iyer, Nagarajan Natarajan, Suresh Parthasarathy, Sriram K. Rajamani, and Rahul Sharma. 2022. Jigsaw: Large Language Models meet Program Synthesis. In *Proc. ACM ICSE*.

[38] Sungmin Kang, Robert Feldt, and Shin Yoo. 2023. Deceiving Humans and Machines Alike: Search-based Test Input Generation for DNNs using Variational Autoencoders. *ACM Transactions on Software Engineering and Methodology* (2023).

[39] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.

[40] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. 2021. Exploiting spatial dimensions of latent in gan for real-time image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 852–861.

[41] Taehun Kim, Kunhee Kim, Joonyeong Lee, Dongmin Cha, Jiho Lee, and Daijin Kim. 2022. Revisiting image pyramid structure for high resolution salient object detection. In *Proceedings of the Asian Conference on Computer Vision*. 108–124.

[42] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4015–4026.

[43] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2019. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems* 32 (2019).

[44] Ang Li, Yichuan Mo, Mingjie Li, and Yisen Wang. 2024. PID: prompt-independent data protection against latent diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*. 28421–28447.

[45] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2023. Seed-bench-2: Benchmarking multimodal large language models. *arXiv preprint arXiv:2311.17092* (2023).

[46] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models. *arXiv preprint arXiv:2407.07895* (2024).

[47] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. 2021. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems* 34 (2021), 16331–16345.

[48] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems* 36 (2024).

[49] Lingbo Liu, Jiaqi Chen, Hefeng Wu, Guanbin Li, Chenglong Li, and Liang Lin. 2021. Cross-modal collaborative representation learning and a large-scale rgbt benchmark for crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4823–4833.

[50] Yunfan Liu, Qi Li, and Zhenan Sun. 2019. Attribute-aware face aging with wavelet-based generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11877–11886.

[51] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[52] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11976–11986.

[53] Mandi Luo, Jie Cao, Xin Ma, Xiaoyu Zhang, and Ran He. 2021. FA-GAN: Face augmentation GAN for deformation-invariant face recognition. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2341–2355.

[54] Farkhod Makhmudkhujaev, Sungeun Hong, and In Kyu Park. 2021. Re-aging gan: Toward personalized face age transformation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3908–3917.

[55] Sondess Missaoui, Simos Gerasimou, and Nicholas Matragkas. 2023. Semantic Data Augmentation for Deep Learning Testing using Generative AI. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 1694–1698.

[56] Qi Pang, Yuanyuan Yuan, and Shuai Wang. 2022. MDPFuzz: testing models solving Markov decision processes. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*. 378–390.

[57] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. In *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17)*. 1–18.

[58] Yun Peng, Shuzheng Gao, Cuiyun Gao, Yintong Huo, and Michael Lyu. 2024. Domain knowledge matters: Improving prompts with fix templates for repairing python type errors. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*. 1–13.

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[60] Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of Pretraining Term Frequencies on Few-Shot Numerical Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 840–854. doi:10.18653/v1/2022.findings-emnlp.59

[61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[62] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. DreamBooth: Fine Tuning Text-to-image Diffusion Models for Subject-Driven Generation. (2023).

[63] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. 2018. Assessing generative models via precision and recall. *Advances in Neural Information Processing Systems* 31 (2018).

[64] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems* 29 (2016), 2234–2242.

[65] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9243–9252.

[66] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. 2023. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089* (2023).

[67] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[68] Yuqiang Sun, Daoyuan Wu, Yue Xue, Han Liu, Haijun Wang, Zhengzi Xu, Xiaofei Xie, and Yang Liu. 2024. GPTScan: Detecting Logic Vulnerabilities in Smart Contracts by Combining GPT with Program Analysis. In *Proc. ACM ICSE*.
[69] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*. 303–314.
[70] Luan Tran, Xi Yin, and Xiaoming Liu. 2017. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1415–1424.
[71] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. 2019. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8198–8207.
[72] Shuai Wang and Zhendong Su. 2020. Metamorphic object insertion for testing object detection systems. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*. 1053–1065.
[73] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023. CogVLM: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079* (2023).
[74] Cheng Wen, Jialun Cao, Jie Su, Zhiwu Xu, Shengchao Qin, Mengda He, Haokun Li, Shing-Chi Cheung, and Cong Tian. 2022. Enchanting Program Specification Synthesis by Large Language Models using Static Analysis and Program Verification. In *Proc. Springer CAV*.
[75] Chunqiu Steven Xia, Matteo Paltenghi, Jia Le Tian, Michael Pradel, and Lingming Zhang. 2024. Universal fuzzing via large language models. *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering* (2024).
[76] Chunqiu Steven Xia, Yuxiang Wei, and Lingming Zhang. 2023. Automated program repair in the era of large pre-trained language models. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 1482–1494.
[77] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. 2019. Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 146–157.
[78] Xiaoyuan Xie, Pengbo Yin, and Songqiang Chen. 2022. Boosting the Revealing of Detected Violations in Deep Learning Testing: A Diversity-Guided Method. In *37th IEEE/ACM International Conference on Automated Software Engineering*. 1–13.
[79] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2021. FILIP: Fine-grained Interactive Language-Image Pre-Training. In *International Conference on Learning Representations*.
[80] Boxi Yu, Zhiqing Zhong, Xinran Qin, Jiayi Yao, Yuancheng Wang, and Pinjia He. 2022. Automated testing of image captioning systems. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*. 467–479.
[81] Yuanyuan Yuan, Qi Pang, and Shuai Wang. 2022. Unveiling Hidden DNN Defects with Decision-Based Metamorphic Testing. In *37th IEEE/ACM International Conference on Automated Software Engineering*. 1–13.
[82] Yuanyuan Yuan, Qi Pang, and Shuai Wang. 2023. Revisiting neuron coverage for dnn testing: A layer-wise and distribution-aware criterion. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 1200–1212.
[83] Yuanyuan Yuan, Qi Pang, and Shuai Wang. 2024. Provably Valid and Diverse Mutations of Real-World Media Data for DNN Testing. *IEEE Transactions on Software Engineering* (2024).
[84] Yuanyuan Yuan, Shuai Wang, Mingyue Jiang, and Tsong Yueh Chen. 2021. Perception Matters: Detecting Perception Failures of VQA Models Using Metamorphic Testing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16908–16917.
[85] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. 2023. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems* 36 (2023), 31428–31449.
[86] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. 2018. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 132–142.
[87] Yuechen Zhang, Jinbo Xing, Eric Lo, and Jiaya Jia. 2024. Real-world image variation by aligning diffusion inversion chain. *Advances in Neural Information Processing Systems* 36 (2024).
[88] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. 2022. RegionCLIP: Region-based Language-Image Pretraining. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 16772–16782.
[89] Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zheng-Jun Zha, Jingren Zhou, and Qifeng Chen. 2021. Low-rank subspaces in gans. *Advances in Neural Information Processing Systems* 34 (2021), 16648–16658.
[90] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-domain gan inversion for real image editing. In *European conference on computer vision*. Springer, 592–608.
[91] Jun-Yan Zhu. 2021. CycleGAN Failure Cases. http://github.com/junyanz/CycleGAN#failure-cases.
[92] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.