

Benchmark Designers Should “Train on the Test Set” to Expose Exploitable Non-Visual Shortcuts

Ellis Brown Jihan Yang Shusheng Yang Rob Fergus Saining Xie

New York University

Abstract

Multimodal Large Language Models (MLLMs) can ace many multimodal benchmarks *without* strong visual understanding, by exploiting linguistic priors, answer patterns, and skewed label distributions. This is particularly problematic for *vision-centric* benchmarks, which explicitly aim to require visual inputs to be solved. If a benchmark *can* be gamed, it *will* be. We take this as the starting principle for robust benchmark design. Extending established partial-input auditing to contemporary multimodal evaluation, we argue that designers should stress-test the exact artifact they release, i.e., directly “train on the test set” to probe the *specific test set* for its intrinsic, exploitable patterns.

We instantiate this as a “Test-set Stress-Test” (TST), applying k -fold cross-validation to text-only diagnostics within the test set. This yields held-out solvability at the benchmark level and a sample-level bias score $s(x)$ that drives “Iterative Bias Pruning” (IBP). The primary diagnostic fine-tunes an LLM with no feature engineering, while a Random Forest variant adds an interpretable designer-side audit. Across VSI-Bench, CV-Bench, MMMU, and VideoMME we find vulnerability to be heterogeneous. Template-based benchmarks show large within-Qwen gains under identical scoring (up to +33 points), while knowledge-intensive MMMU shows essentially none despite high blind scores from pretrained knowledge. Pruning tracks these diagnoses. On VSI-Bench, a designer-in-the-loop pilot cuts the fine-tuned blind score far more than matched random pruning; on MMMU, automated pruning moves the downstream vision-blind gap no more than random removal, as expected where there is no learnable shortcut to remove. TST is a pre-release audit of an evaluation instrument, not a recipe for training deployed models on test answers.

1 Introduction

Despite significant advances in visual understanding, evaluating progress remains challenging. From MNIST [LeCun et al. \(1998\)](#) to ImageNet [Deng et al. \(2009\)](#), from COCO [Lin et al. \(2014\)](#) to VQA [Antol et al. \(2015\)](#), benchmarks have served as the backbone of progress in computer vision. As tasks grew more complex and models more capable, benchmarks shifted from clean, domain-specific datasets to open-ended, real-world evaluations (fig. 1). Today, multimodal LLMs let us ask anything about an image, but this expressive power comes at a hidden cost: *we’ve lost control over what’s being measured*.

The uncomfortable truth is that models can ace multimodal benchmarks without strong visual understanding. They exploit biases, linguistic priors, and superficial patterns. A model can answer a *visual* question *without looking* at the image. It can ground language without grounding perception, yet still achieve a high score. Although concerns about language bias were raised years ago in the VQA era (see section 5), the rise of LLMs as both knowledge engines and sources of shortcuts makes it necessary to reexamine evaluation methodology and benchmark design with greater attention to nuance and complexity [Tong et al. \(2024\)](#).

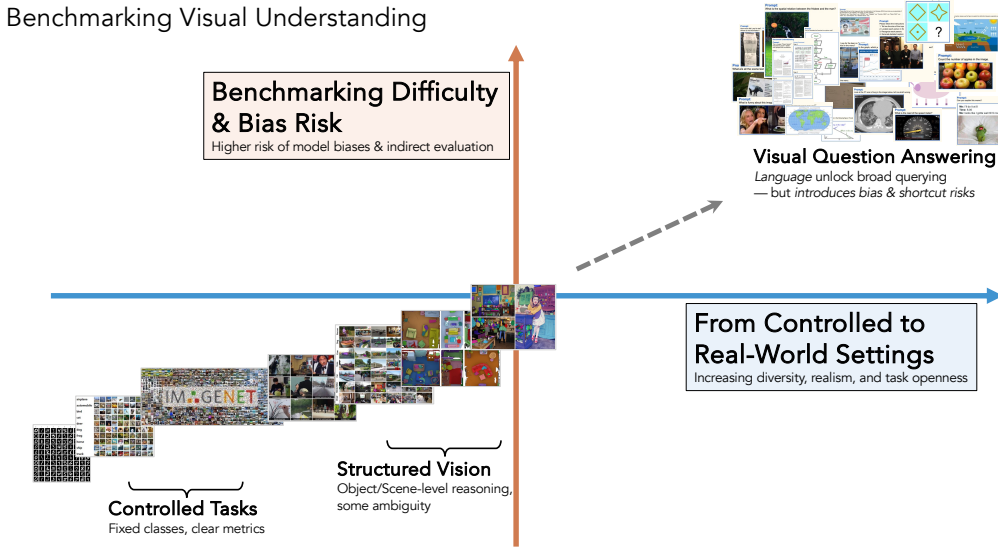


Figure 1: **The Evolving Landscape of Visual Understanding Benchmarks.** As benchmarks evolved from controlled, narrow tasks to open-ended VQA, they gained expressivity but became vulnerable to non-visual shortcuts. Language-driven evaluation enables flexible querying but risks models exploiting linguistic patterns rather than visual understanding.

This observation motivates the central argument of this paper: **multimodal benchmark designers should proactively stress-test their creations for exploitable non-visual shortcuts.** While it has become a common sanity check, the “blind” test Tong et al. (2024) evaluates a *multimodal* benchmark with *vision disabled* and reveals when vision is unnecessary. It offers no insight into *why* specific samples are exploitable or *how* to fix the non-visual shortcuts. We demonstrate that **a rigorous and useful stress test involves directly “training on the test set”¹**, not to overfit, but to adversarially probe the test set for such intrinsic vulnerabilities. This approach elevates robust benchmark design from simple static dataset curation to an iterative and adversarial refinement process.

This “train on the *test set*” framing is critical. While training diagnostic models on held-out in-distribution data can reveal biases that generalize across a domain, such an approach may miss *idiosyncratic vulnerabilities* unique to the *specific test set artifact*, e.g., those arising from the sampling process, templated questions, or human filtering decisions. Our proposed methodology directly targets these test-set-specific shortcuts, helping to ensure that reported benchmark scores reflect genuine multimodal capabilities rather than the exploitation of unique statistical artifacts present in the evaluation instrument.

To demonstrate an effective realization of these principles, we introduce systematic approaches for both diagnosing and mitigating non-visual shortcuts. We first **diagnose** benchmark susceptibility using a “Test-set Stress-Test” (TsT) methodology that applies k -fold cross-validation to train diagnostic models exclusively on non-visual test-set features, deriving both an overall exploitability measure and sample-level bias scores, $s(x)$. We realize this diagnostic through two complementary approaches: a powerful LLM-based variant (TsT-LLM) that LoRA-tunes a language model on question-only inputs to capture complex shortcuts, and a lightweight Random Forest variant (TsT-RF) trained on hand-crafted features that offers efficiency and interpretability for rapid auditing. We then **debias** benchmarks by systematically filtering samples identified as highly biased according to $s(x)$ using an “Iterative Bias Pruning” (IBP) procedure.

Applying our diagnostic framework to prominent *vision-centric* multimodal benchmarks such as VSI-Bench Yang et al. (2025a) and CV-Bench Tong et al. (2024), we uncover sub-

¹This phrasing is a deliberate provocation directed at *benchmark designers*. TsT is a held-out, pre-release audit of an evaluation instrument, not a recipe for training deployed models on test answers.

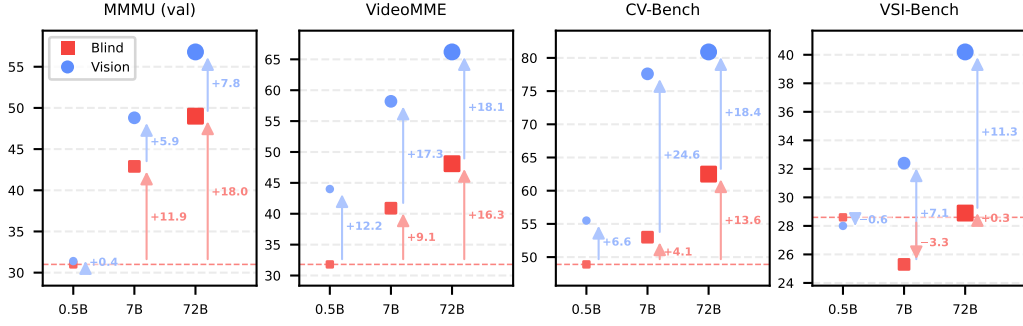


Figure 2: **Knowledge-based shortcuts in multimodal benchmarks.** Blind ■ vs. vision-enabled ● performance across LLaVA-OneVision model scales. MMMU shows substantial gains from scaling the LLM backbone but minimal improvement from enabling vision. VSI-Bench demonstrates the opposite pattern: large vision gains with negligible blind scaling.

stantial non-visual biases, highlighting the prevalence of this issue (section 3). We study benchmark refinement on both VSI-Bench and MMMU: a designer-in-the-loop VSI pilot and separate automated controls expose the method’s provenance and limitations, while MMMU provides a non-template negative control (section 4). Our experiments show that TST reveals test-set-specific distributional patterns on template-based benchmarks that even GPT-4o cannot exploit zero-shot, while correctly identifying minimal learnable shortcuts in knowledge-intensive benchmarks. We further show that IBP reduces non-visual solvability across diverse benchmark types.

In summary, our contributions are fivefold: (1) A principled stance on what constitutes an “exploitable” non-visual shortcut, with a designer-in-the-loop philosophy for responsible debiasing (section 2); (2) A generalizable TST diagnostic framework realized through both a high-efficacy LLM-based variant and an interpretable Random Forest variant (section 3); (3) Empirical validation across four major multimodal benchmarks (VSI-Bench, CV-Bench, MMMU, and VideoMME), revealing pervasive non-visual shortcuts and comparing against frontier LLM baselines (section 3.4); (4) The IBP mitigation methodology with raw scoring, together with random-pruning and diagnostic-robustness controls (section 4); and (5) VSI-Bench-Debiased v1, a designer-in-the-loop pilot released for community use, with its manual provenance made explicit.

In the remainder of this paper, we detail the challenge of non-visual shortcuts (section 2), present our diagnostic and mitigation methodologies (sections 3 and 4), discuss related work (section 5), and conclude with a call for more rigorous benchmark design practices (section 6).

2 The Challenge: Non-Visual Shortcuts

Non-visual shortcuts in multimodal benchmarks take two forms. *Knowledge-based* shortcuts arise when LLMs can answer questions using world knowledge acquired during pretraining without consulting the visual input Tong et al. (2024), for example, answering “How tall is the refrigerator?” from memorized appliance dimensions. fig. 2 makes this concrete across model scales: on MMMU, blind performance rises with the size of the LLM backbone while enabling vision adds little, whereas VSI-Bench shows the opposite pattern (further analysis in section B).

Statistical shortcuts arise from patterns in the test set itself: skewed answer distributions, category-position correlations, and predictable value ranges that allow models to score well by learning distributional regularities rather than performing visual reasoning. As shown in fig. 3, statistical biases manifest across diverse task types. Counting tasks often exhibit severe long-tailed answer distributions; in VSI-Bench, over 50% of counting questions have ground truth “3” or fewer, enabling high scores from frequency priors alone (fig. 3a). Spatial relation

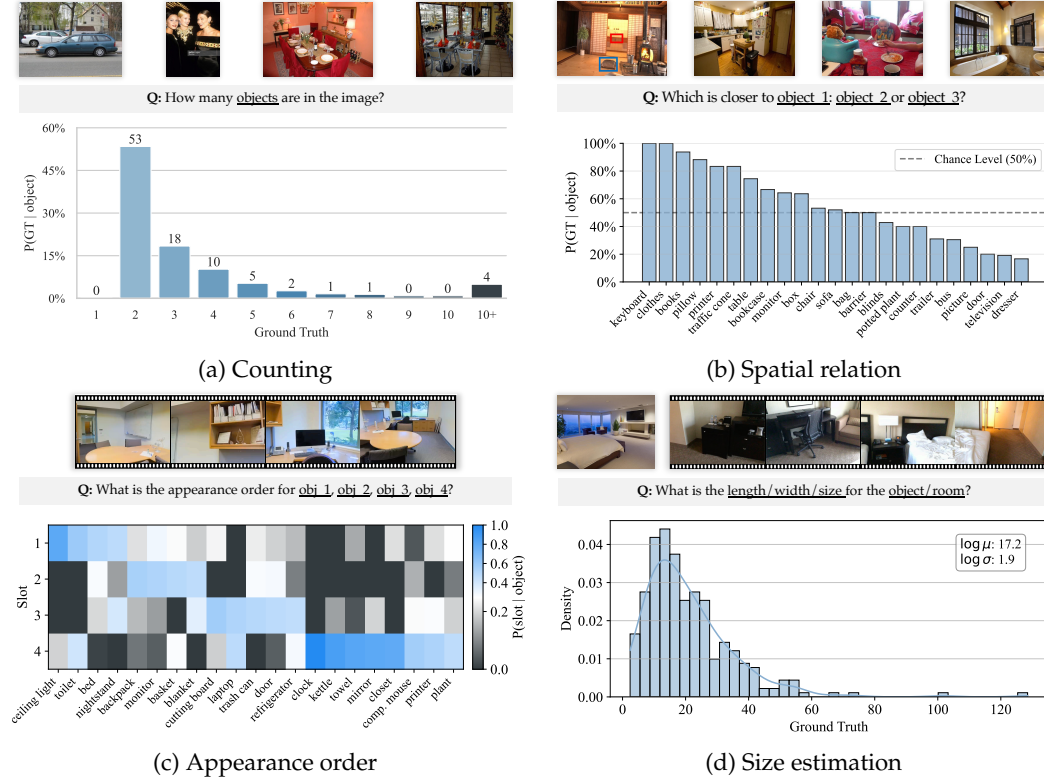


Figure 3: **Statistical biases create non-visual shortcuts across diverse multimodal benchmarks.** (a) Counting tasks exhibit severe long-tailed answer distribution skewness; (b) Spatial relation tasks show imbalanced answer frequencies, where certain object categories appear as correct answers disproportionately often; (c) Appearance order tasks have strong category-position correlations; and (d) Size estimation tasks follow predictable log-normal distributions. Such patterns enable achieving high scores without the visual input.

tasks show imbalanced answer frequencies; in CV-Bench depth, categories like “keyboard” appear as the correct answer 100% of the time they are queried (fig. 3b). Appearance order tasks exhibit strong category-position correlations (fig. 3c), and size estimation tasks follow predictable log-normal distributions (fig. 3d). Both types inflate benchmark scores and misrepresent genuine visual understanding.

Critically, these shortcuts are *readily exploitable*: fine-tuning LLaVA-Video-7B on just 10K in-distribution samples boosts its *blind* VSI-Bench score by +18.8 points, nearly matching the +20.4 point vision-enabled gain. This demonstrates that MLLMs learn statistical shortcuts that bypass visual reasoning entirely (section B, table 3). This motivates a systematic diagnostic methodology that can quantify such vulnerabilities at both benchmark and sample levels.

The interval from the blind zero-shot score (no test-set access) to the TsT score (test-set-specific knowledge) is the benchmark’s *exploit surface*. It is a worst-case stress test, not ordinary model performance. The VSI-Train-10k result shows that this surface is practically relevant: in-distribution training, without access to the test split, already produces a +18.8-point blind gain, close to the +20.4-point vision-enabled gain.

We define an **exploitable non-visual shortcut** as any pattern, regardless of origin, that renders visual input redundant for a task designed to measure visual understanding. TsT is a *diagnostic tool for benchmark designers*: a high bias score $s(x)$ flags a sample as statistically exploitable, but the designer decides the appropriate response (diversify distributions, rewrite questions, or accept the bias). This designer-in-the-loop philosophy ensures debiasing preserves benchmark validity.

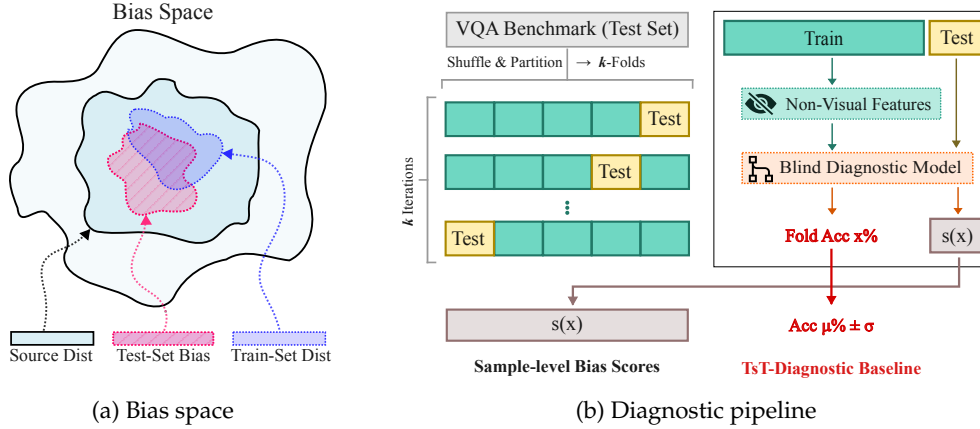


Figure 4: **Test-set Stress-Testing (TsT) targets intrinsic test-set vulnerabilities.** (a) TsT directly probes biases intrinsic to the specific test set (pink region), rather than approximating them via external training data. (b) The test set is split into k folds; a blind diagnostic model trains on $k-1$ folds and evaluates on the held-out fold, repeated k times to yield (i) overall non-visual solvability and (ii) per-sample bias scores $s(x)$.

Responsible use. TsT should be run by benchmark designers before release to audit the benchmark itself. It should not be used to tune deployed models on evaluation answers, and TsT scores should not be reported as ordinary no-access model performance.

3 Diagnosing Non-visual Shortcuts via Test-Set Stress-Testing

Benchmarks can fail in two fundamentally different ways: *training failures*, where biased training data prevents robust generalization (e.g., VQA-CP Agrawal et al. (2018)), and *evaluation failures*, where the test set itself contains exploitable artifacts. Our TsT framework targets the latter. While training on held-out in-distribution data can reveal domain-level biases, it may miss *idiosyncratic vulnerabilities* specific to a particular test set’s composition (fig. 4a). We therefore advocate directly probing the test set via **k -fold cross-validation on non-visual features**, uncovering exploitable biases arising from the specific sampling process, procedural generation logic, or human filtering decisions. We present the core TsT framework (section 3.1), its complementary LLM-based (section 3.2) and RF-based (section 3.3) realizations, and empirical validation across four benchmarks (section 3.4).

3.1 The TsT Framework

The foundational principle of TsT is to **quantitatively estimate the extent to which benchmark questions can be answered using exclusively non-visual information present in the test set itself**. We apply k -fold cross-validation Stone (1974); Hastie et al. (2009) directly on the test set: partition into k disjoint folds (typically $k = 5$), train a diagnostic model on non-visual features from $k-1$ folds, and use the remaining fold as held-out validation. This process repeats k times, so every sample is scored by a model that was never trained on that sample. The aggregate is held-out validation performance, not training accuracy. The TsT framework, illustrated in fig. 4, provides a direct measure of a benchmark’s “non-visual solvability”. The TsT methodology yields two critical outputs:

1. **Overall TsT Score:** The aggregated held-out score across all k folds provides a global estimate of the benchmark’s “non-visual solvability”. A high TsT score indicates that a significant portion can be solved without the visual input by exploiting non-visual patterns, potentially inflating reported MLLM performance. TsT combines two distinct notions of bound: at fixed test-set access, its measured score is a pragmatic lower bound

on what stronger diagnostics might exploit; relative to ordinary no-access evaluation, it deliberately probes a worst-case access regime.

2. **Sample-Level Bias Score $s(x)$:** For each sample x , we define $s(x)$ as the held-out diagnostic’s gold-answer score when x was in the validation fold. High $s(x)$ indicates the sample is likely answerable without visual information, marking it as a candidate for shortcut learning. These scores enable targeted mitigation (section 4).

We reserve $s(x)$ for the held-out fine-tuned gold-answer score used by IBP. For TsT-LLM, $\Delta s(x) = s_{\text{FT}}(x) - s_{\text{ZS}}(x)$ denotes the test-set-learnable component when explicitly reported; TsT-RF has no zero-shot analogue.

3.2 LLM-Based TsT Diagnostic

Our primary diagnostic uses a language model (Qwen2-7B Yang et al. (2024)) to capture learnable patterns ranging from simple statistical correlations to complex textual regularities. For each fold in the k -fold procedure, we fine-tune the LLM using LoRA Hu et al. (2022) on question and answer-choice text (ignoring all visual inputs) from $k-1$ training folds, then predict on the held-out fold. TsT-LLM offers three key advantages: (1) *zero feature engineering*: direct application without manual feature design; (2) *comprehensive detection*: the procedure surfaces both shortcut types, but through two distinct outputs rather than one. The blind zero-shot baseline exposes knowledge-based exploitability the model already carries from pretraining, while Δ_{TsT} isolates the component that is *learnable from the test set itself*; MMMU separates the two cleanly (section 3.4); and (3) *aligned diagnosis*: matching the sophistication of the MLLMs being evaluated. LoRA enables computational efficiency (~ 20 minutes per benchmark on $4 \times \text{H200}$ GPUs for $k=5$ folds), though the method generalizes to full fine-tuning Schulman & Lab (2025). While TsT-LLM provides strong detection capability, it offers limited interpretability regarding *which specific* non-visual cues drive exploitability, motivating the complementary RF-based diagnostic below. Implementation details are in section D.1.

3.3 RF-Based TsT Diagnostic

We complement TsT-LLM with a Random Forest diagnostic (TsT-RF) Breiman (2001) trained on hand-crafted non-visual features (parsed question structure, answer properties, metadata, and task-specific statistics; see section D.3). TsT-RF trades generality for *interpretability* and *efficiency*. Feature importance reveals which cues drive the RF diagnostic (e.g., a single feature achieves 0.968 Gini importance for VSI-Bench size estimation; section F), and training completes in seconds on CPU. Some hand-crafted statistics are label-conditioned quantities available to benchmark designers, so we interpret RF results as designer-side audit diagnostics rather than deployable blind-model performance; label-agnostic feature design is an important direction for future work. TsT-RF is a fast, interpretable secondary diagnostic for template-based benchmarks; TsT-LLM provides the zero-engineering alternative for non-templated data.

3.4 Empirical Validation: TsT Reveals Widespread Shortcut Susceptibility

Applying our TsT diagnostics to four prominent multimodal benchmarks reveals that non-visual shortcut susceptibility is both widespread and significant.

TsT-LLM Results. table 1 presents TsT-LLM diagnostic results across VSI-Bench, CV-Bench, MMMU, and VideoMME. We compare the performance of a blind Qwen2-7B model in zero-shot (ZS) evaluation against its score after k -fold LoRA fine-tuning directly on the test set. The improvement, Δ_{TsT} , quantifies the degree of *learnable* non-visual shortcuts intrinsic to each test set. For template-based benchmarks CV-Bench and VSI-Bench, TsT-LLM achieves dramatic gains of +33.3 and +18.0 points respectively, indicating that substantial fractions of these benchmarks can be “solved” by learning patterns in non-visual data alone. VideoMME shows a meaningful gain of +6.4 points. Notably, on MMMU, a knowledge-intensive and non-templated benchmark, the fine-tuning improvement is negligible ($\Delta_{\text{TsT}} = -0.6$), indicating that MMMU’s text contains very few *learnable* shortcuts beyond what the base model

Table 1: **TsT-LLM reveals learnable non-visual shortcuts distinct from frontier LLM knowledge.** Qwen ZS and TsT use identical scoring, so Δ_{TsT} measures learnable test-set-specific signal. For MC, Qwen/TsT report mean gold-option probability; GPT-4o reports contextual hard top-1 accuracy. All Δ values are computed from unrounded scores.

Benchmark	Blind Zero-Shot			
	GPT-4o hard acc.	Qwen ZS score	TsT score	Δ_{TsT}
CV-Bench	47.3	40.1	73.4	+33.3
VSI-Bench (MC)	25.2	25.5	39.1	+13.6
VSI-Bench (MC+NUM)	–	24.7	42.7	+18.0
MMMU (val)	52.3	34.2	33.6	-0.6
VideoMME	46.7	35.3	41.7	+6.4

VSI MC+NUM weights the MC score and NUM mean relative accuracy by sample count.

VSI NUM only: Qwen ZS 24.0, TsT 46.0 ($\Delta+21.9$); chance is zero.

already knows from pretraining. This contrast between template-based and knowledge-intensive benchmarks demonstrates that TsT effectively discriminates between different types of non-visual vulnerability: substantial distributional exploitability in benchmarks with structured question templates, versus minimal learnable bias in knowledge-heavy evaluations.

Comparison to Frontier LLM Baselines. For context, we also report GPT-4o blind zero-shot performance (table 1).² The primary learnability signal is the within-Qwen ZS-to-TsT change under identical scoring. Large gains on CV-Bench and VSI-Bench indicate learnable test-set-specific patterns; MMMU’s near-zero change, alongside GPT-4o’s high blind hard accuracy, is consistent with pretrained world knowledge instead.

Vision-indispensable control. On MMStar [Chen et al. \(2024a\)](#), the blind zero-shot mean gold-option probability is 24.2%, approximately the 25% uniform baseline, and the TsT score reaches 29.5% ($\Delta_{\text{TsT}} = +5.4$). This is far below VSI-Bench’s +13.6 MC-only gain (+18.0 weighted), providing a curated positive control alongside MMMU’s −0.6 negative control.

TsT-RF Results. TsT-RF reports high non-visual predictability on template-based benchmarks, reaching 43.5% on VSI-Bench and 75.5% on CV-Bench (full results in section D). Because some RF features are label-conditioned (section 3.3), we read these as designer-side audit statistics that corroborate the *direction* of the TsT-LLM finding, not as an independent blind-model measurement. The fact that both diagnostics independently detect substantial bias underscores the severity of the problem. The sample-level bias scores $s(x)$ generated by these diagnostics provide the foundation for targeted mitigation (section 4).

4 Mitigating Non-Visual Shortcuts Guided by TsT Insights

We now demonstrate how TsT-derived bias scores enable systematic benchmark refinement via Iterative Bias Pruning (IBP).

4.1 The Iterative Bias Pruning (IBP) Procedure

IBP applies TsT-derived $s(x)$ scores to iteratively prune the most biased samples, re-diagnosing after each batch to adapt to the evolving data distribution (algorithm 1). This iterative re-computation is crucial: removing highly biased samples alters the statistical landscape of the remaining data, potentially changing the relative exploitability of other samples or the predictive power of different non-visual features. This adaptive approach prevents “shifting the bias under the rug,” where addressing one dominant shortcut leaves secondary ones untouched or amplified. The procedure is governed by a removal budget B

²GPT-4o uses hard accuracy; Qwen/TsT MC uses mean gold-option probability. Absolute values are contextual, not directly comparable.

(controlling the trade-off between bias reduction and dataset coverage), batch size b (influencing re-diagnosis granularity), and an early-stopping threshold τ on maximum residual bias. We use non-difficulty-normalized $s(x)$ scores throughout the mainline analyses; a separately labeled delta-selected MMMU downstream control is reported in section 4.3. We use TsT-RF for the automated VSI analyses (section E.3) and TsT-LLM for MMMU; the framework generalizes to any bias-score source. Further implementation details are in section E.1.

Algorithm 1: Iterative Bias Pruning (IBP)

Input: Dataset $\mathcal{D}$; bias-diagnosis function $\text{ComputeSampleBiasScores}(\cdot)$; removal budget B ; batch size b ; early-stopping threshold τ
Output: Debiased dataset $\mathcal{D}'$

```

 $R \leftarrow 0$ ; // samples removed so far
while  $R < B$  do
   $\{s_i\}_{x_i \in \mathcal{D}} \leftarrow \text{ComputeSampleBiasScores}(\mathcal{D})$ ; // re-diagnose
  if  $\max_i s_i \leq \tau$  then
    break; // early stop: all biases below threshold
   $k \leftarrow \min(b, B - R)$ ;
   $\mathcal{I} \leftarrow \text{SelectBatch}(\{s_i\}_{x_i \in \mathcal{D}}, k, \mathcal{D})$ ; // select batch for removal
   $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{I}$ ;
   $R \leftarrow R + |\mathcal{I}|$ ;
return  $\mathcal{D}' \leftarrow \mathcal{D}$ 

```

4.2 Case Study: VSI-Bench-Debiased v1

VSI-Bench-Debiased v1 is a *designer-in-the-loop manual pilot* that predates the formal automated IBP algorithm. It uses hand-crafted, question-type-specific filtering heuristics and removes 2,768 of 5,130 samples (54.0%). It is therefore evidence for the broader diagnose-and-refine workflow, but it is not the output of Algorithm 1. Automated per-format and per-type results are reported separately in section E.3.

The robust signal is preferential removal of blind-exploitable samples. For the fine-tuned LLaVA-Video model, the blind score drops from 44.7% to 32.0%; the vision-blind gap increases from 12.4 to 16.6 (table 2). An independent re-aggregation of cached per-sample predictions reproduces this same v1 result as 16.7; the 0.1 difference is scoring-pipeline tolerance, not a second artifact. Appendix tables built on that re-aggregation pipeline therefore report 16.7. At matched 54% removal over 10 random seeds, the gap is only 12.4 ± 0.6 ; for Cambrian-S, guided removal gives 26.0 versus 19.4 ± 0.8 for random removal. Thus

Table 2: VSI-Bench-Debiased v1 manual-pilot evaluation. The primary signal is the drop in blind score: for the fine-tuned model, blind falls from 44.7 to 32.0 while vision falls far less (57.1 to 48.7). Equivalently, fine-tuning improves vision and blind nearly equally on the original (+20.4 vs. +18.8) but improves vision much more than blind on the debiased set (+17.4 vs. +11.7), so the debiased set better isolates visual reasoning. The vision-blind gap (Δ_{V-B}) widens accordingly (16.6 vs. 12.4); we treat this widening as a secondary, model-dependent consequence. Gaps are computed from unrounded per-type scores and may differ from the rounded entries by 0.1.

Model Configuration	VSI-Bench (Original)			VSI-Bench-Debiased		
	Vis.	Blind	Δ_{V-B}	Vis.	Blind	Δ_{V-B}
LLaVA-Video 7B (Base)	36.7	25.9	10.8	31.3	20.3	11.0
+ VSI-Train-10k FT	57.1	44.7	12.4	48.7	32.0	16.6
<i>Increase in Δ due to FT</i>	20.4	18.8	1.6	17.4	11.7	5.6
<i>Chance (frequency)</i>		34.0			34.0	

the effect depends on *which* samples are removed, rather than pruning alone. Gap widening is nevertheless model-dependent and is treated as secondary to the direct reduction in blind shortcut score.

Because v1 contains only 2,362 questions and materially changes the question-type distribution, it is a separate evaluation artifact, not a drop-in replacement whose leaderboard scores are directly comparable with the original VSI-Bench. Full per-type retention and random-control results are in section E.3.

4.3 Generalization: IBP on MMMU

To test IBP beyond template-based benchmarks, we first apply raw-score IBP with TsT-LLM to the 900-question MMMU validation pool Yue et al. (2024) (847 multiple-choice and 53 open-ended), removing 100 samples (11.1%). Difficulty normalization is not used. Raw mean score changes from 0.305 to 0.269 and the aggregate TsT score from 32.4% to 28.8%.

Separately, our downstream negative control evaluates a delta-selected subset: delta scoring diagnoses the 847 multiple-choice questions, removes 100, and retains all 53 open-ended questions unchanged. On this 800-question survivor set, LLaVA-OneVision-7B’s vision-blind gap moves only from 6.7 to 7.5 (+0.83), inside the matched 1,000-seed random-pruning band [4.62, 8.75]. The delta-selected items are largely general-knowledge questions rather than systematic visual shortcuts; this control validates the delta-selected variant, not raw-score pruning. Full details are in section E.2.

5 Related Work

VQA Debiasing. Non-visual shortcuts have deep roots in VQA Antol et al. (2015); Goyal et al. (2017), where model-debiasing methods address *training failures* by regularizing models against language priors Agrawal et al. (2018); Ramakrishnan et al. (2018), suppressing unimodal biases Cadene et al. (2019); Liang et al. (2021), or employing counterfactual augmentation Niu et al. (2021); Abbasnejad et al. (2020); Cho et al. (2023). These methods improve model robustness but assume the evaluation benchmark itself is sound. TsT addresses the complementary problem of *evaluation failures* by auditing the test set artifact directly and identifying samples where non-visual patterns render visual input redundant regardless of model training quality.

Benchmark Auditing. TsT extends an established line of partial-input auditing. Hypothesis-only NLI baselines show that a label can often be predicted while ignoring the premise Poliak et al. (2018); adversarial removal methods then mitigate such hypothesis-only bias Belinkov et al. (2019). The structural analogy is direct: the image is the omitted premise, while the question and choices form the retained partial input. We therefore do not claim that partial-input prediction or cross-validation is itself new. Our contribution adapts this principle to contemporary multimodal benchmarks by cross-validating within the intended test artifact, producing held-out sample-level exploitability scores, and using them for iterative benchmark refinement.

In multimodal evaluation, the simplest partial-input diagnostic is the “blind” test Tong et al. (2024), which provides only a dataset-level signal. Other methods quantify modality importance Park et al. (2025), spatial bias Agarwal et al. (2025), and unimodal reliance Chen et al. (2024b). TsT adds sample-level scores for targeted, designer-reviewed refinement; this granularity, rather than the general idea of a partial-input baseline, is the methodological distinction.

Robust Benchmarks. Winoground Thrush et al. (2022), Vinoground Zhang et al. (2024), and the Hateful Memes Challenge Kiela et al. (2020) are designed from the ground up to resist shortcuts. Most relevant to our work, MMStar Chen et al. (2024a) manually curates vision-indispensable samples from existing benchmarks, while MMMU-Pro Yue et al. (2025) augments MMMU with additional answer options. However, manual curation does not scale, and augmentation strategies are benchmark-specific. TsT provides a *quantitative, sample-level* alternative: measuring per-sample exploitability to guide reproducible

refinement of any existing benchmark post-hoc. We note that the artifact we release, VSI-Bench-Debiased v1, is itself a designer-in-the-loop manual pilot, and that our automated VSI selection runs on the designer-side TsT-RF diagnostic; the contribution is a measured, reusable signal to guide curation, not the elimination of designer judgment. A broader shortcut-learning literature studies why deep models prefer available but non-causal features and how such behavior can be interpreted or mitigated [Geirhos et al. \(2020\)](#); [Hermann et al. \(2024\)](#); [Du et al. \(2021\)](#). Visual shortcuts remain important; we focus on non-visual shortcuts because they uniquely violate the stated intent of a vision-centric benchmark whose questions should require looking. A systematic comparison table is in section [G](#).

6 Conclusion

Multimodal benchmarks are the foundation for measuring progress in multimodal AI, yet they remain vulnerable to a critical flaw: models can achieve high scores by exploiting non-visual shortcuts rather than demonstrating genuine visual understanding. This phenomenon risks creating an “illusion of progress” that misdirects research efforts toward pattern matching rather than true multimodal reasoning.

We argue that benchmark designers must proactively “train on the test set,” not to overfit, but to adversarially audit for intrinsic, exploitable patterns. To enable this, we introduce Test-set Stress-Test (TsT), a diagnostic framework that quantifies non-visual exploitability through k -fold cross-validation on test-set text, producing sample-level bias scores $s(x)$ that guide targeted mitigation. Applying TsT to four major benchmarks reveals that non-visual shortcuts are both pervasive and heterogeneous: template-based benchmarks show large within-Qwen gains under identical scoring (up to +33 points), while knowledge-intensive MMMU exhibits near-zero learnable gain despite high blind hard accuracy from frontier models’ pretrained world knowledge. Our results separate a promising designer-in-the-loop pilot from automated Iterative Bias Pruning (IBP). VSI-Bench-Debiased v1 reduces blind shortcut scores and widens the fine-tuned vision-blind gap beyond matched random pruning, while automated raw-score IBP reduces intrinsic bias scores. On MMMU, IBP behaves like random removal downstream, the desired negative-control outcome for a benchmark with near-zero learnable shortcuts.

Rigorous test-set stress-testing should become standard practice in the robust design of multimodal benchmarks. Several promising directions remain: extending TsT to modalities beyond vision-language; using TsT-RF signals to guide reparative strategies such as question rewriting, answer rebalancing, and diversification rather than removal; and studying transfer across diagnostics. For open-ended tasks, a candidate signal is improvement in a blind fine-tuned LM’s ability to *produce* the gold answer, paired with a label-shuffle or marginal/format control; raw gold-answer perplexity alone can be confounded by answer marginals. Embedding-space proximity to retrieved test-set neighbors is another option. We release VSI-Bench-Debiased v1 and report exact diagnostic configurations in section [D.1](#) to support careful adoption; the complete public code migration is ongoing. Only through adversarial evaluation of our evaluation instruments can we ensure benchmarks measure genuine multimodal understanding rather than statistical pattern matching.

References

- Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *CVPR*, 2020.
- Amit Agarwal, Hitesh Laxmichand Patel, Srikant Panda, Hansa Meghwani, Jyotika Singh, Karan Dua, Paul Li, Tao Sheng, Sujith Ravi, and Dan Roth. RCI: A score for evaluating global and local reasoning in multimodal benchmarks. In *EMNLP (Industry Track)*, pp. 138–157, 2025. doi: 10.18653/v1/2025.emnlp-industry.10.
- Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don’t just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.
- Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *NeurIPS*, 2021.
- Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. On adversarial removal of hypothesis-only bias in natural language inference. In **SEM*, pp. 256–262. Association for Computational Linguistics, 2019. doi: 10.18653/v1/S19-1028. URL <https://aclanthology.org/S19-1028/>.
- Leo Breiman. Random forests. *Machine learning*, 2001.
- Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing unimodal biases for visual question answering. *NeurIPS*, 2019.
- Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, pp. 27056–27087, 2024a. doi: 10.52202/079017-0850.
- Meiqi Chen, Yixin Cao, Yan Zhang, and Chaochao Lu. Quantifying and mitigating unimodal biases in multimodal large language models: A causal perspective. In *Findings of EMNLP*, pp. 16449–16469, 2024b. doi: 10.18653/v1/2024.findings-emnlp.960.
- Jae Won Cho, Dong-Jin Kim, Hyeonggon Ryu, and In So Kweon. Generative bias for robust visual question answering. In *CVPR*, 2023.
- Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. Towards interpreting and mitigating shortcut learning behavior of nlu models. In *NAACL*, pp. 915–929, 2021. doi: 10.18653/v1/2021.naacl-main.71.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

- Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2 edition, 2009. doi: 10.1007/978-0-387-84858-7.
- Katherine L Hermann, Hossein Mobahi, Thomas Fel, and Michael C Mozer. On the foundations of shortcut learning. In *ICLR*, 2024. URL <https://arxiv.org/abs/2310.16228>.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022.
- Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *NeurIPS*, 2020.
- Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
- Zujie Liang, Haifeng Hu, and Jiaying Zhu. Lpf: A language-prior feedback objective function for de-biased visual question answering. In *SIGIR*, 2021.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, 2024.
- Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *CVPR*, 2021.
- Jean Park, Kuk Jin Jang, Basam Alasaly, Sriharsha Mopidevi, Andrew Zolensky, Eric Eaton, Insup Lee, and Kevin Johnson. Assessing modality bias in video question answering benchmarks with multimodal large language models. In *AAAI*, 2025.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In **SEM*, pp. 180–191. Association for Computational Linguistics, 2018. doi: 10.18653/v1/S18-2023. URL <https://aclanthology.org/S18-2023/>.
- Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *NeurIPS*, 2018.
- John Schulman and Thinking Machines Lab. LoRA Without Regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. <https://thinkingmachines.ai/blog/lora/>.
- Mervyn Stone. Cross-validators choice and assessment of statistical predictions. *Journal of the royal statistical society*, 1974.
- Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022.
- Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs. *NeurIPS*, 2024.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. doi: 10.48550/arXiv.2407.10671. URL <https://arxiv.org/abs/2407.10671>.

- Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. *CVPR*, 2025a.
- Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Daohan Lu, Rob Fergus, Yann LeCun, Li Fei-Fei, and Saining Xie. Cambrian-S: Towards Spatial Supersensing in Video. *arXiv preprint arXiv:2511.04670*, 2025b.
- Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, 2024.
- Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhui Chen, and Graham Neubig. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In *ACL*, pp. 15134–15186, 2025. doi: 10.18653/v1/2025.acl-long.736.
- Jianrui Zhang, Mu Cai, and Yong Jae Lee. Vinoground: Scrutinizing lmms over dense temporal reasoning with short videos. *arXiv preprint arXiv:2410.02763*, 2024.

Acknowledgments

We thank Shengbang Tong and Anjali Gupta for reviewing this manuscript, Chris Hoang for helpful discussions, and Zaid Khan for advice on LoRA frameworks. E.B. is supported by the DoD NDSEG Fellowship Program. S.X. acknowledges support from the MSIT IITP grant (RS-2024-00457882) and the NSF award IIS-2443404.

Appendix

This appendix provides extended analyses, moved content from the main paper, and implementation details: §A LLM-use disclosure; §B extended shortcut analysis; §C VSI-Train-10k generation; §D TsT diagnostic details; §E debiasing details; §F RF interpretability; §G extended related work.

A LLM Disclosure

LLMs were used for drafting and editing assistance, bibliographic and consistency checks, coding assistance, and data visualization during preparation of this manuscript. The authors directed the work, verified the reported claims, citations, and numerical results against source artifacts, and take responsibility for the final text. Research ideas, experimental design, data generation, and evaluation were performed and validated by the authors.

B Extended Shortcut Analysis

B.1 What Constitutes a Non-Visual Shortcut?

We define an **exploitable non-visual shortcut** as any pattern, regardless of origin, that renders visual input redundant for a task designed to measure visual understanding. A pattern becomes problematic when it lets models bypass the intended visual reasoning, whether it reflects natural world knowledge, real-world statistics, or procedural generation artifacts.

This definition has important implications. Consider a visual size estimation task asking “How tall is the refrigerator?” A model can answer from memorized appliance dimensions without visual measurement. Similarly, statistical correlations (e.g., “lamp” being disproportionately correct in spatial relation questions) function as shortcuts regardless of whether they reflect real-world regularities or sampling artifacts.

B.2 Knowledge-Based Shortcuts

The first category of exploitable shortcuts arises from world knowledge embedded in LLMs during pretraining Tong et al. (2024). fig. 2 reports this across LLaVA-OneVision scales: MMMU and VideoMME benefit more from scaling the LLM backbone than from enabling visual inputs, while VSI-Bench shows negligible gains from LLM scaling in blind settings but substantial improvements when vision is enabled.

B.3 Statistical Shortcuts

As illustrated in fig. 3, statistical biases manifest across diverse task types. Counting tasks exhibit severe long-tailed distributions; spatial relation tasks show imbalanced answer frequencies; appearance order tasks have category-position correlations; and size estimation tasks follow predictable log-normal distributions. These persist even in benchmarks designed to avoid knowledge-based shortcuts.

Table 3: **MLLMs readily exploit statistical shortcuts.** LLaVA-Video-7B on VSI-Bench before/after fine-tuning on VSI-Train-10k. Blind score gains (+18.8 pts) nearly match vision-enabled gains (+20.4 pts).

Configuration	Vision	Blind	Δ_{V-B}
LLaVA-Video 7B (Base)	36.7	25.9	10.8
+ VSI-Train-10k FT	57.1	44.7	12.4
Δ due to FT	+20.4	+18.8	+1.6
Chance (frequency)		34.0	

B.4 MLLMs Readily Exploit These Shortcuts

Fine-tuning LLaVA-Video-7B on just 10K in-distribution samples (table 3) boosts the blind score by +18.8 points, nearly matching the +20.4 point vision-enabled gain. This demonstrates that MLLMs learn statistical shortcuts that bypass visual reasoning entirely.

C VSI-Train-10k Generation

To create the in-distribution training set for demonstrating MLLM shortcut learning (table 3), we follow the VSI-Bench Yang et al. (2025a) benchmark curation pipeline. First, we extract object numbers, bounding boxes, and room sizes from the *training splits* of ScanNet Dai et al. (2017), ScanNet++ Yeshwanth et al. (2023), and arkitScenes Baruch et al. (2021). We then use the exact same question templates from VSI-Bench to generate question-answer pairs for seven rule-based question types (excluding route planning). Finally, we randomly sample 10K questions, setting a maximum of 20 questions per question type per scene. See Yang et al. (2025b) for more details on the procedure followed.

D TsT Diagnostic Details

D.1 TsT-LLM Implementation Details

For our LLM-based TsT diagnostic (section 3.2), we use Qwen2-7B-Instruct Yang et al. (2024) (revision f2826a0) and VSI-Bench revision d7cb1a3. The exact configuration that produced every headline result is: $k=5$, seed 42, one repeat, LoRA Hu et al. (2022) rank $r=8$, $\alpha=16$, dropout 0.1, all-module targets, one epoch per fold, train and evaluation batch size 4, gradient accumulation 1, learning rate 2×10^{-4} with cosine decay and 0.1 warmup, FP16, maximum sequence length 1024, a 10% internal training-fold validation split, and the gemma training template. Inference uses raw prompts (no chat template), temperature 0, top- p 1, at most 10 new tokens, and top-100 log probabilities. MC samples are scored with the probability of the gold option, while NUM samples use generated answers and mean relative accuracy; difficulty normalization is off. Each sample is scored only in its held-out fold.

An independent reproduction of the primary configuration yields a 42.66% weighted VSI score ($\Delta_{TsT}=17.93$), matching the reported 42.7% ($\Delta_{TsT}=18.0$). Rank 128 is not recommended in this small-data regime.

For completeness, we also report a separate ranking-oriented configuration ($r=32$, $\alpha=64$, two epochs, batch size 32, learning rate 5×10^{-5} with cosine decay and 0.1 warmup, qwen template with chat formatting). Within that study, a matched $r=128$, $\alpha=256$, three-epoch run used the same chat-formatted setup; their un-normalized $\Delta s(x)$ rankings have Spearman $\rho=0.87$ and 70.9% top-1,000 overlap. This comparison is specific to the chat-formatted study and does not establish cost or ranking agreement relative to the primary configuration; the alternate should be treated as a distinct ranking recipe rather than a lower-cost substitute.

D.2 TsT-RF Results

Table 4: **TsT-RF provides efficient, interpretable diagnostics.** Random Forest performance via k -fold cross-validation on hand-crafted features. Some features are label-conditioned designer-side statistics (section 3.3), so the aggregate CV score is an audit diagnostic and is not directly comparable to a blind model’s score; the baseline columns are listed for scale only. VSI-Bench combines MC accuracy and NUM mean relative accuracy (MRA), while CV-Bench uses MC accuracy.

Benchmark	Chance baseline	Frequency baseline	TsT-RF CV score
CV-Bench	33.3	43.1	75.5
VSI-Bench	-	34.0	43.5

D.3 TsT-RF Feature Extraction

For RF-based TsT diagnostic (section 3.3), we extract non-visual features $f_{nv}(x)$ tailored to each benchmark’s structure. For template-based benchmarks (VSI-Bench, CV-Bench), features include: (1) entities and relations parsed from question text; (2) answer-space properties such as answer format and option count; (3) question-type and object metadata; and (4) task-specific frequency and distributional statistics (e.g., for size estimation, the average size of mentioned objects computed from dataset statistics). For non-templated benchmarks, the RF feature set is restricted to available question/answer properties and question-level metadata. The reported implementation uses scikit-learn forests with no maximum depth: 100 trees for classifiers and 200 for regressors. A robustness grid over $n_{\text{estimators}} \in \{200, 1000, 2000\}$ and $\text{depth} \in \{10, 20, \text{None}\}$, applied to both model types, brackets this setting from above rather than containing it, since the shipped forests use fewer trees. Across that grid, pairwise raw-score Spearman correlation is at least 0.990 and top-1,000 removed-set Jaccard is at least 0.873, so the rankings are insensitive to forest size and depth over an order of magnitude in $n_{\text{estimators}}$. See section D.4 for more details on the feature engineering process.

D.4 TsT-RF Feature Engineering

The following tables summarize representative non-visual features engineered for each benchmark’s template-based question types. The implementation additionally derives closely related frequency and ratio features; these tables are illustrative rather than an exhaustive schema. The features capture textual, statistical, and metadata patterns that might correlate with ground truth answers without requiring visual input.

VSI-Bench Features. Based on the VSI-Bench question templates and dataset structure, we extract task-specific features for each question type; representative features are shown in table 5. VSI-Bench uses templated questions with predictable structure, enabling efficient feature extraction via regular expressions that parse object categories, distances, directions, and other task-specific elements directly from the question text.

CV-Bench Features. Based on the CV-Bench question templates and task structure, we extract features for each visual reasoning task, shown in table 7. All frequency maps and aggregate statistics are fit exclusively on the training folds during k -fold cross-validation.

Table 5: **Representative VSI-Bench TsT-RF features by task type.** Features are extracted from the question text and answer choices (for Multiple-Choice) or the question alone (for Open-Ended), with no visual input. All statistical features are computed exclusively from training folds during cross-validation.

Task	Format	Feature Type	Feature Name	Description
Object Counting	OE	Categorical	object	Object category
		Numerical	obj_count	Count of this object
			obj_val_log_mean	Mean of log GT values
			obj_val_log_std	Std dev of log GT values
			global_mean_log	Global mean (log space)
Object Size Estimation	OE	Categorical	object	Object category
		Numerical	obj_count	Count of this object
			obj_val_log_mean	Mean of log GT sizes
			obj_val_log_std	Std dev of log GT sizes
Object Abs. Distance	OE	Categorical	object_pair	Sorted pair of objects
		Numerical	pair_count	Count of this pair
			pair_val_mean_log	Mean of log distances
			pair_val_std_log	Std dev of log distances
Room Size Estimation	OE	Numerical	global_mean_log	Global mean (log space)
			global_std_log	Global std dev (log space)

Table 6: **Representative VSI-Bench TsT-RF features by task type (continued).**

Task	Format	Feature Type	Feature Name	Description
Object Rel. Distance	MC	Categorical	target_object object_{i}	Category of target object Object category of option i , $\forall i \in [4]$
		Numerical	opt_{i}.obj_freq	Frequency of option i 's object being correct
			opt_{i}.pair_freq	Frequency of option i 's object-target pair being correct
			max_opt_obj_freq max_opt_pair_freq	Max option frequency Max object-target pair frequency
Object Rel. Direction	MC	Categorical	difficulty positioning_object orienting_object	Subtype (easy/med/hard) Object that is being stood by Object that is being faced toward
			querying_object	Query target object
		Numerical	obj_freq_score	Combined frequency of all 3 objects being in a question
Route Planning	MC	Categorical	beginning_object facing_object target_object opt_{i}	Starting position object Initial orientation object Destination object Route string option i
		Numerical	num_steps	Number of steps in route options (all the same)
			num_choices opt_{i}.freq_score	Number of MC options Route option i frequency
			obj_freq_score	Combined frequency of all 3 objects being in a question
Appearance Order	MC	Categorical	opt_seq_{i}	Object sequence for option i : $[o_1^i, o_2^i, \dots, o_n^i]$
		Numerical	seq_{i}.pos_score	Sum of frequencies of object o_j^i appearing in position j in a GT sequence, $\forall j \in [n]$
			seq_{i}.adj_pair_score	Sum of frequencies of adjacent pairs o_j^i, o_{j+1}^i in a GT sequence, $\forall j \in [n-1]$
			seq_{i}.comb_pair_score	Sum of frequencies of all combinatorial pairs o_j^i, o_k^i in a GT sequence, $\forall j, k \in [n], j < k$
			seq_{i}.score	Average of the above three scores for option i

Table 7: **Representative CV-Bench TsT-RF features by task type.** Features are extracted from the question text and answer choices with no visual input. All CV-Bench tasks are Multiple-Choice. Statistical features are computed exclusively from training folds during cross-validation.

Task	Feature Type	Feature Name	Description
2D Count	Categorical	object	Object category
	Numerical	n.options	Number of MC choices
		obj_count	Count of this object
		obj_val_log_mean	Mean of log GT values
		obj_val_log_std	Std dev of log GT values
		global_mean_log	Global mean (log space)
		global_std_log	Global std dev (log space)
		opt_{i}.dist_from_obj_mean opt_{i}.dist_from_global_mean	Per-option distance scores Per-option distance scores
2D Spatial Relation	Categorical	object_{i}	Category of object i , $\forall i \in [2]$
	Numerical	n.options	Number of MC choices
		pair_freq_score	Frequency of object pair in questions
		question_length	Question character length
		contains_{dir}	Contains spatial keyword dir , for $dir \in \{\text{left, right, above, below, front, behind}\}$
3D Depth	spatial_keyword_count		Total spatial direction keywords
	Categorical	object_{i}	Category of object i , $\forall i \in [2]$
	Numerical	n.options pair_freq_score	Number of MC choices Frequency of object pair in questions
3D Distance	Categorical	object_{i}	Category of object i , $\forall i \in [2]$
	Numerical	n.options pair_freq_score	Number of MC choices Frequency of object pair in questions

E Debiasing Details

The automated VSI-Bench analyses use label-conditioned TsT-RF statistics available to benchmark designers. They therefore characterize designer-side audit selections rather than predictions by a deployable blind model.

E.1 Iterative Bias Pruning (IBP) Details

The IBP algorithm, presented in algorithm 1, operationalizes this iterative debiasing philosophy. It begins with the full dataset $\mathcal{D}$ and progressively refines it. In each iteration, current bias scores $\{s_i\}_{x_i \in \mathcal{D}}$ are computed for all remaining samples using the `ComputeSampleBiasScores( $\cdot$ )` function (which encapsulates our TsT methodology). Based on these scores, and respecting an overall removal budget B and a per-iteration batch size b , a subset of samples $\mathcal{I}$ is selected for removal by the `SelectBatch( $\cdot$ )` function. Selection can remove the highest-scoring samples directly, weight numerical samples, or balance categorical groups, while always using $s(x)$ as its primary signal.

The process repeats until the budget B is exhausted or an early-stopping criterion, such as the maximum remaining bias score $\max_i s_i$ falling below a threshold τ , is met.

The budget B controls the trade-off between bias reduction and dataset coverage, while batch size b influences granularity: smaller b enables more frequent re-diagnosis at higher computational cost. While we focus on pruning due to its simplicity and reproducibility, the framework could accommodate reparative strategies (question rewriting, answer rebalancing) in future work.

E.2 MMMU Negative-Control Results

Raw-score IBP over the 900-question validation pool removes 100 samples: mean $s(x)$ falls from 0.305 to 0.269 and the aggregate TsT score from 32.4% to 28.8%. Separately, the downstream evaluation below uses the delta-selected survivor set. That variant diagnoses the 847 multiple-choice questions, removes 100, and leaves all 53 open-ended questions unchanged.

Table 8: **The MMMU delta-selected subset is a downstream negative control.** Delta-guided removal changes the gap by only 0.83 points, inside the 1,000-seed matched-random band.

Subset	n	Vision	Blind	Gap
Original	900	49.4	42.8	6.7
Delta-guided removal	800	47.9	40.4	7.5
Random removal	800	–	–	6.69 ± 0.58

Random gap range: [4.62, 8.75]; guided gap 7.50.

E.3 VSI-Bench-Debiased v1 Provenance and Diversity

The released VSI-Bench-Debiased v1 is the original designer-in-the-loop pilot, not the output of automated IBP. It uses bespoke per-question-type heuristics and hard-coded budgets to remove 2,768 of 5,130 questions. All ten types remain, but coverage is not uniformly preserved: for example, 85.7% of medium and 68.9% of hard relative-direction questions are removed, while only 2.3% of easy relative-direction questions are removed. This distribution shift is a limitation of v1 and motivates a reproducible automated v2.

Table 9: **Actual v1 manual-pilot composition.** Coverage remains across all types, but retention changes materially.

Question Type	Format	Original	Removed	Removed (%)	Kept
Object size	NUM	953	600	62.9	353
Absolute distance	NUM	834	400	48.0	434
Relative distance	MC	710	400	56.3	310
Appearance order	MC	618	300	48.5	318
Object counting	NUM	565	314	55.6	251
Relative direction (med.)	MC	378	324	85.7	54
Relative direction (hard)	MC	373	257	68.9	116
Relative direction (easy)	MC	217	5	2.3	212
Room size	NUM	288	88	30.6	200
Route planning	MC	194	80	41.2	114
Total		5,130	2,768	54.0	2,362

E.3.1 Manual and automated results.

Automated per-format IBP ranks MC and NUM separately because their score scales are not commensurate. At $B=1000$ it removes 19.5% and reduces intrinsic scores, but the LLaVA gap remains 12.3. Automated per-type variants at matched 54% removal widen the gap to 14.3–15.4, depending on allocation and seed; the hand-crafted v1 pilot reaches 16.7. These are separate artifacts and should not be conflated.

Table 10: **VSI refinement provenance.** Gaps use the paper’s micro aggregation and the fine-tuned LLaVA-Video model, re-aggregated from cached per-sample predictions; this pipeline reproduces table 2’s 16.6 as 16.7.

Selection method	Removed (%)	Kept	Vision-blind gap
Original	0.0	5,130	12.4
Automated per-format, $B=1000$	19.5	4,130	12.3
Automated per-type, uniform	54.0	2,360	14.3
Automated per-type, allocated (seed 1/42)	54.0	2,362	14.4/15.4
Manual pilot v1	54.0	2,362	16.7

Table 11: **Matched random-pruning controls** at v1’s 54% removal rate (10 seeds). Gaps come from the same re-aggregation pipeline as table 10, hence 16.7 rather than 16.6.

Model	Full gap	Guided v1	Random
LLaVA-Video-7B + FT	12.4	16.7	12.4 ± 0.6
Cambrian-S	19.2	26.0	19.4 ± 0.8

Table 12: **VSI-Bench-Debiased v1 transfer across model families.** Micro-averaged scores on the original and released v1 manual-pilot sets. V and B denote vision and blind evaluation; each combines MC accuracy and NUM MRA. Gaps and Δ gap are computed from unrounded scores, so they may differ from the rounded entries by 0.1.

Model	LLM backbone	Original			v1			Δ gap
		V	B	Gap	V	B	Gap	
Cambrian-S	Qwen2.5	66.1	46.9	19.2	58.3	32.3	26.0	+6.8
LLaVA-Video-7B + FT	Qwen2	57.1	44.7	12.4	48.7	32.0	16.7	+4.3
InternVL3-9B	InternLM2	43.6	32.5	11.1	41.5	28.2	13.2	+2.2
InternVL2.5-26B	InternLM2.5	45.3	32.5	12.8	38.4	25.1	13.3	+0.5
InternVL2.5-8B	InternLM2.5	44.0	31.6	12.4	37.5	24.2	13.3	+0.9
LLaVA-OneVision-7B	Qwen2	35.9	25.6	10.3	29.6	19.0	10.6	+0.2
LLaVA-Video-7B	Qwen2	37.0	23.6	13.5	31.9	18.6	13.3	-0.2

InternVL3-9B is a fresh, verified non-Qwen run (InternLM2 backbone); its gap increases from 11.1 to 13.2, about three standard deviations above its matched random-pruning null. Cambrian-S is Qwen2.5-backed and is not used as the non-Qwen claim. Across all seven evaluated models, gap widening tracks original blind exploitability (Spearman $\rho=0.96$), while the original gap does not ($\rho=0.14$); this descriptive trend explains why weak blind exploiters remain nearly flat.

E.3.2 Diagnostic dependence and robustness.

TsT-LLM and TsT-RF identify complementary shortcut families. Their removed sets have Jaccard 0.057 (65 shared of 1,135 in the union), so the framework is diagnostic-aware rather than defining a unique ground-truth pruned set.

Table 13: **Removal rates (%) for two diagnostics.** Budgets differ (LLM 200, RF 1,000); matched-budget Jaccard remains 0.058.

Question type	TsT-LLM	TsT-RF
Appearance order	15.5	40.6
Object counting	15.9	13.8
Object size	0.4	30.5
Absolute distance	0.2	14.0

Swapping Qwen2-7B for LLaMA-3-8B under matched folds and seed gives raw-score Spearman 0.58, Pearson 0.61, and 59.7% top-1,000 overlap. Within TsT-RF, an estimator/depth grid gives pairwise Spearman at least 0.990 and top-1,000 Jaccard at least 0.873. Thus model families are complementary, while each diagnostic’s ranking has meaningful within-family robustness.

E.3.3 Removal-budget sensitivity.

For automated per-format IBP, mean bias decreases monotonically with B . MC mean bias begins to plateau around $B=1500$ (29.2% removal), while NUM mean bias continues to decline through the tested range. We use $B=1000$ as a coverage/bias trade-off; this sweep concerns intrinsic bias reduction, not v1’s separate gap-widening pilot.

Table 14: **Automated per-format IBP budget sweep.**

B	Removed (%)	MC mean bias	NUM mean bias
200	3.9	0.335	0.421
500	9.7	0.314	0.384
1,000	19.5	0.275	0.322
1,500	29.2	0.254	0.262
2,000	39.0	0.225	0.199
2,500	48.7	0.213	0.133

F TsT-RF Interpretability Analysis

To demonstrate how TsT-RF can supply actionable dataset diagnostics, we present a case study of VSI-Bench’s `object_size_estimation` task. RF feature importance explains the RF diagnostic’s dependence on engineered dataset features; it is not a faithful explanation of the internal mechanism used by a modern MLLM.

Task and Features. This task asks: “What is the length of the longest dimension (length, width, or height) of the {object}, measured in centimeters?” For each question, we extract six features: one categorical (object), and five numerical features derived from the training-fold data (`obj_count`, `obj_freq_score`, `obj_val_log_mean`, `obj_val_log_std`, and `obj_val_log_ratio` for the object category). All statistical features are computed exclusively from the training folds during k -fold cross-validation, so no validation sample informs its own score. These statistics remain label-conditioned by construction (section 3.3), which is precisely what makes them a designer-side audit signal rather than a blind-model performance estimate.

Diagnostic Performance. Applying TsT-RF via 5-fold cross-validation, the diagnostic model achieved an overall MRA of $61.4\% \pm 2.4\%$, substantially above the frequency baseline of 34.0%, indicating strong non-visual exploitability.

Table 15: Gini importances of the six RF features used for the VSI-Bench size-estimation case study.

Feature	Importance
<code>obj_val_log_mean</code>	0.968
object (category)	0.009
<code>obj_val_log_std</code>	0.008
<code>obj_val_log_ratio</code>	0.007
<code>obj_count</code>	0.004
<code>obj_freq_score</code>	0.004

Feature Importance Analysis. table 15 presents the Gini feature importances. The `obj_val_log_mean` feature (object category’s average size) dominates with importance 0.968, while all other features combined contribute less than 0.04. This reveals that the RF diagnostic essentially ignores question structure, object frequency, and size variance, instead simply memorizing and predicting the typical size of each object category.

High- $s(x)$ Examples and Root Cause. table 16 shows the top-ranked samples by bias score $s(x)$. All involve object categories with extremely low size variation in the dataset: dishwashers ($90 \pm 3\text{cm}$, coefficient of variation 0.037), beds ($216 \pm 17\text{cm}$, CV 0.080), and washers ($87 \pm 5\text{cm}$, CV 0.058). For comparison, objects with high size variation like ceiling lights ($72 \pm 56\text{cm}$, CV 0.778) or radiators ($146 \pm 110\text{cm}$, CV 0.755) received much lower $s(x)$ scores.

Actionable Insight for Benchmark Design. This analysis provides a concrete, quantitative prescription for improving the benchmark: the diagnostic model learned to exploit the fact that certain object categories have near-constant sizes, allowing it to bypass visual estimation entirely. Benchmark designers can address this by either (1) removing questions about low-variance objects, or (2) ensuring that sampled instances of each object category exhibit diverse sizes. This exemplifies how TsT-RF translates statistical patterns into specific, implementable design improvements, complementing TsT-LLM’s detection of complex shortcuts.

Table 16: High- $s(x)$ examples and corresponding object size statistics from VSI-Bench size estimation.

Object	Mean Size (cm)	Std Dev (cm)	Coef. of Var.	Rank by $s(x)$
Dishwasher	90.4	3.4	0.037	1st
Bed	216.1	17.2	0.080	2nd
Washer	87.1	5.0	0.058	3rd
Kettle	23.8	1.5	0.062	4th
Mouse	11.6	1.1	0.091	5th
<i>Low-variance objects (easily exploitable)</i>				
Ceiling Light	71.8	55.8	0.778	Low
Radiator	146.3	110.4	0.755	Low

G Extended Related Work

G.1 VQA Debiasing

The challenge of non-visual shortcuts has deep roots in VQA [Antol et al. \(2015\)](#); [Goyal et al. \(2017\)](#). VQA-CP [Agrawal et al. \(2018\)](#) deliberately introduces distribution shift to expose models relying on question-type priors. Subsequent training-time interventions include adversarial regularization [Agrawal et al. \(2018\)](#); [Ramakrishnan et al. \(2018\)](#), unimodal bias suppression (RUBi [Cadene et al. \(2019\)](#)), feedback-based objectives [Liang et al. \(2021\)](#), counterfactual augmentation [Niu et al. \(2021\)](#); [Abbasnejad et al. \(2020\)](#), and generative bias modeling [Cho et al. \(2023\)](#). These methods improve model robustness but assume the evaluation benchmark itself is sound. TsT addresses the complementary problem: auditing the test set artifact directly.

G.2 Modern Benchmark Auditing

Beyond the coarse “blind” test [Tong et al. \(2024\)](#), recent diagnostic tools include the Modality Importance Score (MIS) [Park et al. \(2025\)](#) for modality ablation in video QA, the Region Comprehension Index (RCI) [Agarwal et al. \(2025\)](#) for spatial bias analysis, and causal frameworks for unimodal bias quantification [Chen et al. \(2024b\)](#). TsT distinguishes itself by quantifying *learnable* non-visual patterns intrinsic to the test set through k -fold cross-validation, producing sample-level bias scores $s(x)$ that enable targeted refinement.

Table 17: TsT complements existing approaches by auditing test-set intrinsic artifacts.

Method	Goal	Target	Methodology	Output
VQA-CP Agrawal et al. (2018)	Model Robustness	Train Priors	Train-Test Shift	OOD Acc.
RUBi Cadene et al. (2019)	Model Debiasing	Model Behavior	Bias Suppression	Debiased Model
Counterfactual Niu et al. (2021)	Model Debiasing	Model Behavior	Data Synthesis	Debiased Model
GenB Cho et al. (2023)	Model Debiasing	Model Behavior	Generative Modeling	Debiased Model
MIS Park et al. (2025)	Benchmark Audit	Modality Use	Modality Ablation	Importance Score
RCI Agarwal et al. (2025)	Benchmark Audit	Spatial Bias	Patch-based Eval	RCI Score
TsT (Ours)	Benchmark Audit	Test Set Artifacts	k-fold CV on Text	$s(x)$, Aggregate Score

G.3 Benchmarks Designed for Robustness

Winoground [Thrush et al. \(2022\)](#), Vinoground [Zhang et al. \(2024\)](#), and the Hateful Memes Challenge [Kiela et al. \(2020\)](#) are designed from the ground up to resist shortcuts. MM-Bench [Liu et al. \(2024\)](#) aims for robustness through comprehensive skill coverage. Most relevant, MMStar [Chen et al. \(2024a\)](#) manually curates vision-indispensable samples, and MMMU-Pro [Yue et al. \(2025\)](#) augments MMMU with additional options. TsT provides an automated, quantitative alternative applicable to any existing benchmark post-hoc.