Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

FrameShift-CAD: Executable Coordinate-Frame Interventions for Diagnosing Text-to-CAD Generation

Shengyao Sun

sthfornothing@sjtu.edu.cn

Shanghai Jiao Tong University https://orcid.org/0009-0008-9175-8226

Research Article

Keywords: text-to-CAD, coordinate frames, spatial reasoning, metamorphic testing, geometric transformations, benchmark evaluation, large language models

Posted Date: September 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10904095/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

FrameShift-CAD: Executable Coordinate-Frame Interventions for Diagnosing Text-to-CAD Generation

Shengyao Sun | SJTU Paris Elite Institute of Technology (SPEIT), Shanghai Jiao Tong University

Correspondence: sthfornothing@sjtu.edu.cn | ORCID: 0009-0008-9175-8226

Preprint. This manuscript has not undergone peer review.

Abstract

Text-to-CAD systems are commonly evaluated on independently generated objects, which makes it difficult to determine whether a model can transport the same design through a specified coordinate operation. We present FrameShift-CAD, a controlled diagnostic extension of a deterministic primitive-based CAD benchmark. FrameShift-CAD derives matched active-object and passive-frame prompts from the same base specification and compares each generated shape set with programmatically constructed correct, inverse, unchanged, partial, component-only, and wrong-order counterfactuals. An initial 18-family study across three model families established a translation-rotation gap and motivated a prospectively frozen confirmation over 12 CAD families, two stable deployments, paired active/passive conditions, three pure rotations, and a non-commutative rotation-then-translation operation. In the confirmatory intent-to-treat analysis, exact translation was correct in 32/48 outputs (66.7%), whereas pure rotation was correct in 12/144 (8.3%), a difference of 0.583 (95% family-cluster bootstrap confidence interval, 0.410-0.743). The gap persisted for 90-degree rotations about the X and Y axes and a 120-degree diagonal-axis rotation; all three pooled contrasts had Holm-adjusted $p = 0.0029$. Among count-matched pure-rotation outputs, 99/128 (77.3%) were closest to the unchanged reference, while only 5/128 (3.9%) were closest to the inverse. Compound outputs were fully correct in 9/44 cases, and 20/27 uniquely classified non-correct outputs omitted a component or applied the components in the wrong order. Prompt controls indicate that coordinate-frame terminology increases difficulty, but an explicit mapping does not reliably eliminate the rotation deficit. The results establish an operation-specific reliability gap and provide a reusable method for diagnosing geometric transformation execution.

Keywords: text-to-CAD; coordinate frames; spatial reasoning; metamorphic testing; geometric transformations; benchmark evaluation; large language models

Highlights

• Matched CAD prompts isolate translation, rotation, frame convention, and operation order.

• Prospective confirmation finds a 58.3-point translation-rotation correctness gap.

• Rotation errors are predominantly unchanged-oracle outputs, not inverse transforms.

• Compound instructions frequently preserve translation while dropping rotation.

• Prompt controls separate terminology burden from failure to execute an explicit map.

Introduction

Language-conditioned CAD generation has moved from free-form mesh production toward structured, editable representations. Recent systems generate parametric command sequences, code-like CAD programs, or direct geometric primitives from natural-language descriptions (Khan et al., 2024); (Li et al., 2025); (Guan et al., 2025); (Pyatov et al., 2026). This shift makes numerical spatial reasoning a central reliability requirement. A generated program can be syntactically valid and visually plausible while placing parts at incorrect coordinates, orienting a sketch on the wrong plane, or failing to preserve a design under a requested change. CAD-GPT, for example, identifies starting position and orientation as persistent difficulties and introduces an explicit spatial mechanism to address them (Wang et al., 2024).

Evaluation has expanded accordingly. C2CAD-Bench isolates spatial reasoning by asking models to return deterministic primitive arrays and scoring them against parametric targets (C2CAD-Bench Research Team, 2026). Text2CAD-Bench broadens task complexity and application diversity (Wang et al., 2026), while CADTestBench replaces single-reference similarity with executable requirement tests over generated B-reps (Mallis et al., 2026). CADEngBench goes further by testing functional edits, manufacturability-related checks, physics simulation, joint-frame prediction, and kinematic verification (Singh et al., 2026). These benchmarks measure increasingly important aspects of CAD quality. They do not, however, directly answer whether a generator that receives a design specification plus an exact coordinate operation actually executes that operation.

The distinction is consequential because a static score conflates different failure modes. Suppose a specification is followed by an instruction to rotate the entire object. A low final score could mean that the model applied the inverse rotation, preserved the original coordinates, transformed only some parts, changed the wrong fields, or failed to generate the base design. Those behaviors require different remedies. Coordinate-aware verification and repair are already active topics: CADLoop reports near-miss failures involving deterministic mappings, including coordinate-frame transforms and workplane alignment, and repairs generated programs with symbolic geometric operations (Zhang et al., 2026). FrameShift-CAD builds controlled interventions that quantify which operations fail and identify how they fail.

FrameShift-CAD addresses this measurement problem. It treats a base CAD specification as the source case, appends a frozen coordinate-operation block, and constructs exact local counterfactuals from the public target geometry. Active conditions move the physical object in a fixed world frame. Passive conditions leave the object fixed and express it in the inverse-transformed coordinate frame. Because both variants are generated from the same source, they separate the requested operation from the design content. A field-aware distance then compares a model output with multiple candidate outcomes rather than collapsing all error into one score.

The study follows a staged discovery-to-confirmation design. An offline measurement gate first established that the transformations and counterfactuals were identifiable. An 18-family matrix then tested a fixed Z-axis rotation and a fixed three-dimensional translation on Qwen, DeepSeek, and MiMo deployments. A prompt-control experiment varied whether frame terminology and explicit coordinate formulae were present. Finally, a prospectively frozen experiment replaced the original operation with a fresh translation, rotations about the X and Y axes, a diagonal-axis rotation, and a rotation-then-translation composition. The confirmatory analysis used family-cluster resampling, intent-to-treat handling of unavailable or malformed outputs, a fixed tie threshold, and an independently implemented C2CAD score as sensitivity analysis.

The paper makes three contributions. First, it specifies a controlled transformation protocol for structured text-to-CAD generation, including paired active/passive prompts and schema-preserving affine operations. Second, it provides a counterfactual oracle taxonomy that distinguishes correct, inverse, unchanged, partial, component-only, and wrong-order outputs. Third, it presents evidence that the tested deployments execute exact translations substantially more often than rotations across multiple axes and angles, with unchanged-oracle behavior and loss of the rotation component dominating the error structure. Together, these elements turn coordinate transport into an executable diagnostic for structured CAD generation. Searches of arXiv, CVF Open Access, OpenReview, official project pages, and benchmark repositories through 2 September 2026 located no earlier text-to-CAD study combining paired active/passive interventions, a prospective translation-versus-multi-axis-rotation contrast, a non-commutative compound test, and the same executable error decomposition.

The analysis addresses three questions:

1. Under matched, machine-checkable interventions, how much more often do the tested deployments execute exact translations than exact three-dimensional rotations?

2. Does this operation gap persist across active and passive descriptions, multiple rotation axes and angles, and a compound operation?

3. Which counterfactual error classes dominate, and do prompt controls support a purely terminological explanation?

Related work

Language-conditioned parametric CAD generation

Text-to-CAD research spans learned command-sequence generation, language-model adaptation, code generation, and multimodal synthesis. Text2CAD maps prompts ranging from novice descriptions to detailed geometric instructions into sequential parametric models (Khan et al., 2024). CAD-Llama translates CAD sequences into a structured code-like representation and adapts a pretrained language model through CAD-specific pretraining and instruction tuning (Li et al., 2025). CAD-Coder instead generates CadQuery scripts and optimizes geometric and formatting rewards (Guan et al., 2025). CADFS scales this direction with a large program corpus and a common framework for text-conditioned generation and reconstruction (Pyatov et al., 2026).

These systems differ in representation and training, but all must map linguistic constraints to geometric parameters. CAD-GPT makes this dependence explicit by separately encoding spatial position, sketch-plane rotation, and planar coordinates (Wang et al., 2024). Such mechanisms motivate operation-level evaluation. A model that performs well on familiar sketch-extrude distributions may still fail when every position and direction vector must be transported consistently through a rigid transformation. FrameShift-CAD complements these systems with a controlled method for testing that transport behavior at the output level.

CAD benchmarks, executable checks, and geometric verification

C2CAD-Bench is the direct foundation of the present study. It removes the code layer and requires a JSON array of geometric primitives, allowing deterministic comparison with parametric targets across 25 families and four phases of difficulty (C2CAD-Bench Research Team, 2026). FrameShift-CAD reuses its public Level-1 cases, primitive conventions, and official Geometry scorer. The experimental unit and primary question are different: the present work generates matched follow-up specifications, executes known affine relations over the target arrays, and scores transformation-specific alternatives.

Other 2026 benchmarks occupy complementary parts of the evaluation space. Text2CAD-Bench provides 600 curated tasks spanning simple geometry, complex topology, free-form surfaces, and non-mechanical applications (Wang et al., 2026). CADTestBench evaluates whether generated B-reps satisfy prompt-linked geometric and topological predicates, with mutation analysis used to calibrate the tests (Mallis et al., 2026). CADEngBench evaluates whether parts remain valid under parameter changes and whether assemblies and simulations satisfy engineering criteria (Singh et al., 2026). These works establish that general benchmark novelty, executable testing, and functional CAD evaluation are not new claims available to FrameShift-CAD.

CADLoop is the closest work at the level of coordinate-aware failure. It separates structural inference from deterministic execution and repairs continuous parameters in near-miss Fusion 360 programs (Zhang et al., 2026). Its motivating failures include coordinate-frame and workplane alignment. RA-CAD similarly uses execution feedback, but learns a generate-execute-critique-rewrite policy rather than measuring paired transformation behavior (Yan et al., 2026). FrameShift-CAD contributes systematic measurement: matched controls, an operation contrast, prospective cross-axis confirmation, and counterfactual error attribution.

Transformation diagnostics and metamorphic evaluation

The intervention design is related to metamorphic testing, which derives follow-up cases connected to source cases by necessary relations (Chen et al., 1998). Recent work has applied this logic broadly to LLM evaluation; a 2026 systematic survey synthesizes 93 studies across reliability, code, retrieval, dialogue, and agentic systems (Zheng et al., 2026). FrameShift-CAD uses a stronger oracle than many metamorphic tests because the transformation of the structured target

can be computed exactly. The metamorphic relation defines the paired prompt and expected transport, while deterministic counterfactuals diagnose the resulting output.

Transformation perturbations are also established in spatial-reasoning evaluation. SGP-Bench applies translations and rotations to SVG and CAD programs and measures the consistency of semantic answers (Qiu et al., 2025). SpinBench separates translation, rotation, relative pose, and viewpoint change in visual-language reasoning and reports pronounced rotational weakness (Zhang et al., 2025). FoR-T2I constructs matched camera-view and object-frame prompts for image generation and finds substantially lower accuracy under object-relative descriptions (Gu et al., 2026). Those studies prevent broad claims that translation-rotation comparison or paired frame-of-reference testing is new. They also clarify the remaining gap: none evaluates generation of complete structured CAD outputs under paired three-dimensional active/passive maps with exact correct, inverse, unchanged, partial, and operation-order counterfactuals.

Methods

Study design and experimental stages

FrameShift-CAD used six numbered stages. M0 was an offline measurement-validation gate. M1 tested six Level-1 families and established the initial operation signal. M2 added the remaining 12 Level-1 families for two stable deployments. M3 introduced prompt controls on the six M1 families. M4 attempted to establish an additional independent deployment path but did not yield a stable, scored matrix and was excluded from capability conclusions. M5 evaluated a complete 18-family matrix on a MiMo deployment. M6 was the prospectively frozen transformation-breadth confirmation.

The final evidence chain contains three related but distinct datasets. The first has 18 families, five conditions, and three model families (270 cells). The second adds 96 prompt-control cells and reuses 48 frozen M1 transformed cells rather than querying them again. The third contains 12 families, ten transformed conditions, and two stable deployments (240 cells). No M6 outcome was available when its cases, transformations, primary estimator, tie threshold, bootstrap procedure, or decision rule were frozen.

Base cases and family selection

Source specifications and parametric targets came from the public C2CAD-Bench release (C2CAD-Bench Research Team, 2026). The discovery matrix used one Level-1 case from each of 18 families. M1 selected six cases only from public target geometry and pre-model identifiability criteria, with coverage of all four benchmark phases. M2 added every Level-1 family not present in M1. M5 used the same 18-family matrix.

M6 selected 12 cases without consulting M1-M5 model scores. Eligibility required exact expressibility under all ten conditions, active/passive round-trip validity, counterfactual identifiability, and a target-size estimate below the 32,768-token completion ceiling. The frozen 18-family pool contained only two phase-3 cases, so an infeasible provisional three-per-phase allocation was amended before model execution to 4/2/2/4 cases from phases 1-4. Within phases, selection used shape-count spacing and a fixed SHA-256 tie order. The resulting sample covered seven primitive types (beam, box, cone, cylinder, pipe, sphere, and torus) and expected shape counts from 7 to 72.

The Level-1 restriction was intentional. It kept the base geometry analytically transparent while varying primitive type, assembly pattern, and set size. This design strengthens internal identification of the coordinate operation but does not represent industrial CAD complexity.

Active and passive transformations

Let a primitive position be p \in R^3, a direction vector be v, a rotation matrix be R, and a translation be d. An active object transformation maps

p' = Rp + d, v' = Rv.

The matched passive condition reports the unchanged physical object in the inverse-transformed frame:

p' = R^T(p-d), v' = R^Tv.

Position fields included `center`, `start`, `end`, `base_center`, and `tip_center` plus documented aliases. Direction fields included `axis` and its aliases. Signed-permutation rotations were used so that axis-aligned boxes remained exactly expressible: their size vector was permuted by |R|. Scalar radii, heights, and widths were unchanged. All target transformations were computed locally; targets and oracle labels were never included in an API request.

M1-M5 used a 90-degree Z-axis rotation and translation d_1=(13,-7,5) mm. M6 deliberately replaced these with a fresh translation d_2=(-11,8,6) mm and three pure rotations:

• R_x(90^deg): (x,y,z) -> (x,-z,y);

• R_y(90^deg): (x,y,z) -> (z,y,-x);

• R_(1,1,1)(120^deg): (x,y,z) -> (z,x,y).

M6 also tested the active composition p'=R_z(90^deg)p+d_2. Its passive counterpart used the exact affine inverse. Because rotation and translation do not commute, the alternative order generated a distinct wrong-order oracle.

Prompt construction and controls

Every request contained the unmodified base specification followed by a visually separated `FRAME OPERATION` block. The system message required one JSON object with exactly one top-level `shapes` array, prohibited Markdown and explanations, specified millimetres, and instructed the model not to redesign, simplify, reorder, or add geometry. The standard transformed prompt combined semantic wording with the exact coordinate map. For example, an active rotation stated that the world frame remained fixed, the complete physical object was rotated, and the displayed mapping had to be applied to all affected fields exactly once. The corresponding passive prompt stated that the object remained fixed while its coordinates were expressed in the changed frame.

M3 separated two parts of this instruction. The **formula-only** variant removed the terms *active* and *passive* and retained only the exact field mapping. The **terminology-only** variant described the physical object or frame operation but omitted the coordinate formula. The original term-plus-formula prompts were reused as a frozen baseline. Thus, M3 tested whether terminology burden alone explained the gap and whether an explicit mapping alone was sufficient for reliable execution.

Deployments and generation protocol

The stable discovery deployments were `qwen-plus` through Alibaba Model Studio and `deepseek-v4-pro` through the TokenRhythm gateway. M5 added `mimo-v2.5-pro` through TokenRhythm. M6 returned the exact concrete DeepSeek identifier `deepseek-v4-pro-0813` in all successful canary responses; a provenance-only amendment accepted that single alias before the remaining requests were sent. Results are attributed to dated deployments rather than assumed immutable foundation-model versions.

Requests used the OpenAI-compatible chat interface, server-sent-event streaming, `temperature=0`, `top_p=1`, JSON-object response mode, and `max_completion_tokens=32768`. Thinking was disabled where the deployment supported it. Each model-family-condition cell was sampled once. There was no capability retry, JSON repair, code-fence stripping, arithmetic evaluation, or manual correction. Transport and HTTP failures were recorded as availability outcomes. The model received only the public specification and operation block; target geometry and all counterfactuals remained local.

The initial GLM route produced only 15 HTTP-200 responses in 30 M1 requests and substantial provider-side reasoning usage. M4 likewise failed to establish a complete stable route. Neither was treated as negative capability evidence. MiMo was retained because 87/90 requests returned HTTP 200, although its strict-object coverage was lower than that of Qwen

and DeepSeek. This protocol-level distinction prevents endpoint availability from being interpreted as geometric incapability.

Canonicalization, alignment, and counterfactual oracles

Model outputs were canonicalized using documented primitive and field aliases. For example, `cube` and `cuboid` mapped to `box`; `position`, `origin`, and `location` mapped to `center`; and dictionary vectors with `x`, `y`, and `z` keys were converted to numeric triples. Shapes were aligned by unique identifiers when both output and target contained the same identifier set. Otherwise, a deterministic greedy alignment minimized primitive-type mismatch and centroid distance. Missing or extra shapes incurred a fixed penalty.

For every transformed condition, the local evaluator constructed four base counterfactuals:

1. **correct:** the requested active or passive transform;

2. **inverse:** the opposite-direction transform;

3. **unchanged:** the untransformed public target;

4. **partial:** the correct transform applied to alternating target shapes.

Compound conditions added **rotation-only**, **translation-only**, and **wrong-order** counterfactuals. The partial oracle is a diagnostic reference, not a complete taxonomy of all possible partial transformations.

The primary relation distance evaluated only fields affected by the requested operation. For an aligned position field, Euclidean error was normalized by max(| g|_2,1), where g is the target vector in millimetres. Direction-vector error used Euclidean distance directly. Rotated box-size error used the analogous normalized Euclidean distance. A missing affected field or a missing/extra shape contributed a penalty of 2. Field errors were averaged within a shape and then across aligned shapes and count penalties.

An output was assigned to an oracle only when that oracle was the unique nearest candidate under the frozen tie rule. At threshold epsilon, every oracle whose distance was within epsilon of the minimum entered the winner set; a multi-oracle set was classified as a tie. The primary threshold was epsilon=0.01, with 0, 0.001, and 0.05 prespecified as sensitivity values. The label “unchanged” therefore means closest to the untransformed target, not literal identity with a separately sampled base output.

Measurement validation

M0 evaluated the complete transformation and classification pipeline before the main matrices were frozen. Active/passive round trips passed in 48/48 tests over observed canonicalized outputs. Across 18 candidate Level-1 cases, four-oracle synthetic calibration correctly recovered 1,728/1,728 labels at epsilon\leq0.01 and 1,724/1,728 at epsilon=0.05. Perturbations covered exact targets, reordered aliases, numeric strings, a dropped shape, and 1% and 5% coordinate noise.

The M6-specific gate checked all ten affine conditions. All 180 case-condition cells were schema-expressible, all 180 active/passive round trips passed, and every counterfactual set was self-identifiable. Synthetic recovery at epsilon=0.01 was 99.68%. The official C2CAD scorer uniquely separated 150/180 cells, which motivated its use as a restricted sensitivity analysis rather than the primary diagnostic.

Outcomes and statistical analysis

The primary M6 estimand was the intent-to-treat correct-rate difference between the fresh translation and the pooled pure rotations:

\Delta = P(correct\midtranslation)-

P(correct\midpure rotation).

Transport failures, HTTP errors, malformed objects, and shape-count mismatches were not geometrically reclassified and counted as not correct in intent-to-treat analysis. Two sensitivity populations were also reported: strict parsed objects regardless of count, and strict objects with the expected shape count. Availability, parsing, and count completeness remained separate protocol outcomes.

Confidence intervals used 20,000 family-cluster bootstrap resamples with root seed 20260901. A family was resampled with all of its active/passive and operation cells. Pooled analysis retained deployment strata and averaged deployment-specific family effects before resampling. Each of the three rotation-specific contrasts used a two-sided exact sign-flip test over family differences; Holm adjustment was applied within each model and in the pooled analysis. M3 used the same 20,000-draw family-cluster procedure but was interpreted as a model-by-prompt-variant control experiment rather than confirmatory proof of mechanism.

The official C2CAD Geometry score served as an external sensitivity measure on prevalidated identifiable contrasts. For each output, the oracle with the largest official Geometry score was selected, with ties retained. Additional sensitivity analyses varied epsilon, excluded shape-count mismatches, and conditioned on strict parsing. Active/passive differences and the representation-by-operation-by-deployment cells were prespecified secondary summaries.

Reproducibility and integrity controls

Every manifest froze request order, payload hash, local target hash, model and provider label, generation parameters, and estimated completion bounds. Every response record preserved status, returned model identifier, timing, usage, finish reason, parse outcome, and raw-response SHA-256. The 240 M6 requests completed without retry or repair. The scored-row artifact has SHA-256 `548643fc9dfa40887f687f697a80d0cbc7e4710f50b51ffdf8497c506983aaaf`. A deterministic rerun reproduced transformation scoring, official-score sensitivity, bootstrap intervals, and sign-flip/Holm tests. Sixteen local unit tests covering normalization, affine transforms, counterfactual construction, and statistical logic passed at manuscript assembly.

Results

Protocol coverage

The 18-family, three-deployment matrix contained 270 authorized requests. Qwen returned HTTP 200 for 85/90 requests and a strict `shapes` object for 84. DeepSeek returned HTTP 200 and a strict object for 81/90. MiMo returned HTTP 200 for 87/90 but a strict object for 61. The requested shape count was present in 73 Qwen, 76 DeepSeek, and 55 MiMo outputs. Across deployments, 253/270 requests returned HTTP 200, 226/270 yielded a strict object, and 204/270 matched the expected count. This matrix consumed 613,276 recorded model tokens.

M3 completed all 96 newly authorized requests. Each stable deployment returned HTTP 200 and a strict object for 46/48 requests. The four unavailable cells were preserved as protocol outcomes. M3 consumed 205,207 tokens.

M6 completed all 240 frozen requests and consumed 527,018 tokens. Qwen returned 117/120 HTTP-200 responses; every successful response was a strict object, and 108 had the expected shape count. DeepSeek returned 120/120 HTTP-200 responses, 113 strict objects, and 103 count-matched objects. There were three Qwen transport failures, seven DeepSeek parse failures, and no length truncations. No main-stage output was repaired or retried.

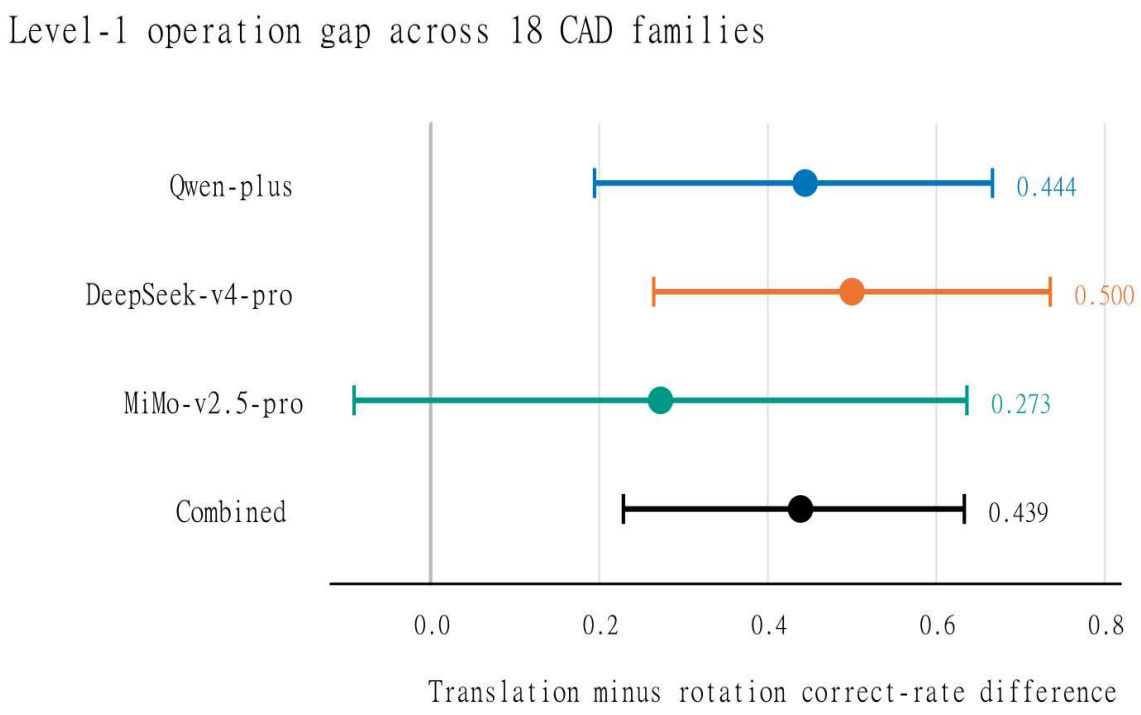
The 18-family matrix established a translation-rotation gap

All three deployments showed higher correct-oracle rates for translation than for rotation (Figure 1; Table 1). Qwen produced 22 correct translations among 33 strictly parsed transformed outputs and 8 correct rotations among 33, a family-cluster difference of 0.444 (95% CI, 0.194-0.667). DeepSeek produced 23/29 correct translations and 10/34 correct rotations, a difference of 0.500 (95% CI, 0.265-0.735). MiMo produced 9/20 correct translations and 4/26 correct rotations,

a difference of 0.273 (95% CI, -0.091-0.636). The MiMo point estimate reproduced the direction, but its interval crossed zero and its strict-output coverage was lower.

Pooling the deployments descriptively, translation was correct in 54/82 strictly parsed outputs (65.8%) and rotation in 22/93 (23.7%). The family-cluster difference was 0.439 (95% CI, 0.229-0.633). When all authorized transformed cells were retained and unavailable or unparsable outputs were counted as not correct, the pooled difference remained 0.296 (95% CI, 0.111-0.482). The operation contrast was therefore not created by conditioning exclusively on successful endpoint responses.

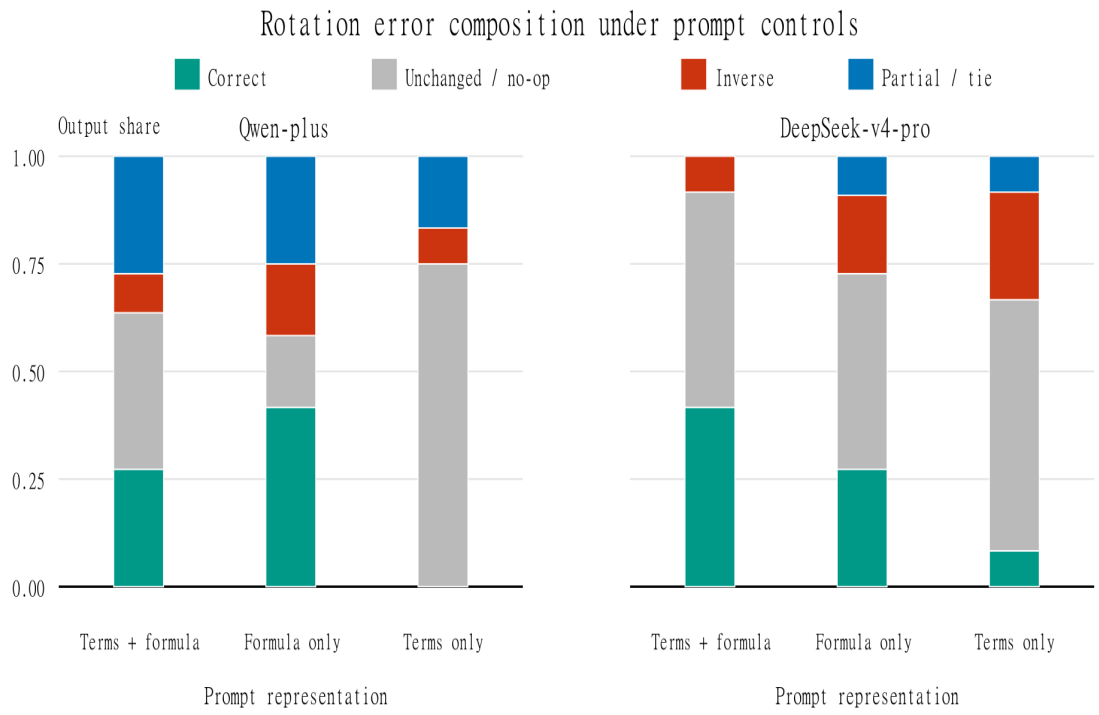
The official C2CAD sensitivity analysis agreed with the primary relation-oracle label on 74/89 outputs (83.2%) for which both metrics produced a unique class. The documented-alias adapter eliminated evaluator exceptions without inferring undocumented geometry. The direction of the official-score operation contrast matched the primary result.



Rotation errors were no-op-like rather than inverse-dominant

The initial taxonomy contradicted the hypothesis that the dominant failure would be selection of the wrong active/passive sign. Among 93 strictly parsed rotation outputs across Qwen, DeepSeek, and MiMo, 41 were closest to the unchanged target and only five were closest to the inverse; the remaining outputs included correct, partial, and tied classifications (Figure 3). The ordering held within each deployment. Qwen produced 14 unchanged and one inverse classification, DeepSeek 14 and three, and MiMo 13 and one.

An unchanged-oracle classification places the transformed output closest, on operation-affected fields, to the untransformed public target. The pattern is no-op-like under-execution, whereas inverse-frame errors were rare.

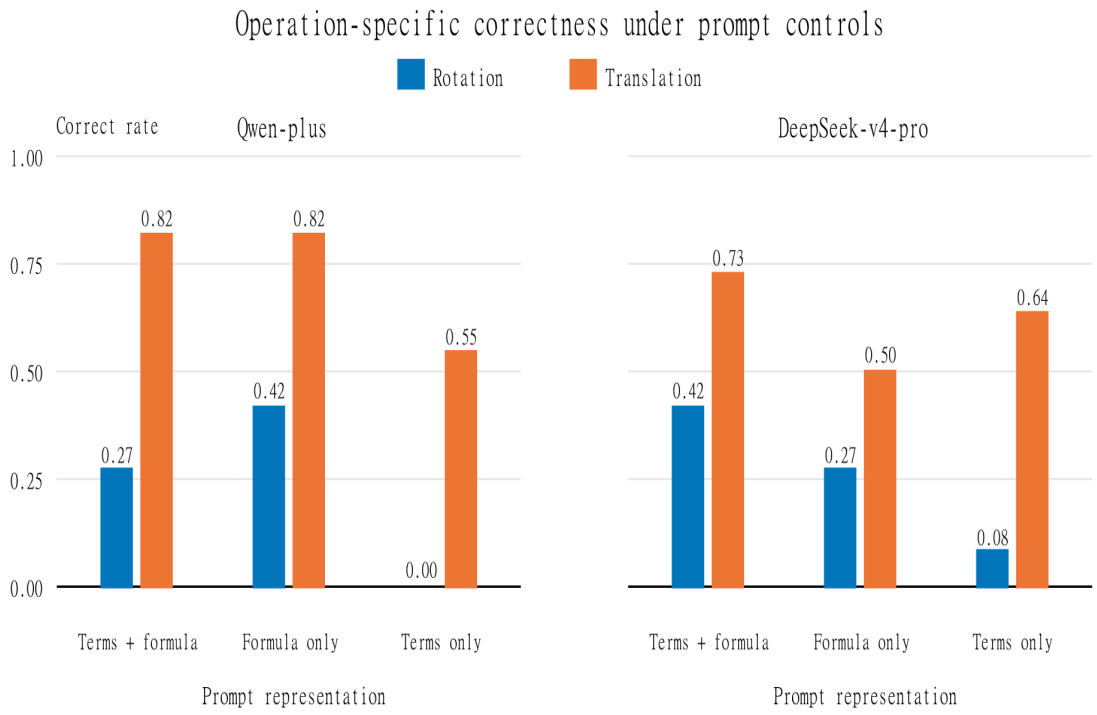


Prompt controls showed a model-dependent formula effect

M3 compared term-plus-formula, formula-only, and terminology-only prompts over six families and four transformed conditions (Figure 2; Table 2). Removing frame terminology did not eliminate the operation gap. For Qwen, formula-only prompts increased rotation correctness from 3/11 to 5/12 while translation remained 9/11; the difference decreased from 0.583 (95% CI, 0.333-0.833) to 0.417 (95% CI, -0.083-0.917). For DeepSeek, rotation decreased from 5/12 to 3/11 and translation from 8/11 to 6/12; the formula-only difference was 0.250 (95% CI, -0.167-0.667). The opposing rotation changes, +0.144 for Qwen and -0.144 for DeepSeek, rule out a common formula-only improvement claim.

Terminology-only prompts produced the lowest rotation correctness in both deployments. Qwen had 0/12 correct rotations and 6/11 correct translations, a difference of 0.500 (95% CI, 0.250-0.750). DeepSeek had 1/12 and 7/11, respectively, a difference of 0.583 (95% CI, 0.500-0.750). Unchanged classifications accounted for 9/12 Qwen rotations and 7/12 DeepSeek rotations, whereas inverse classifications accounted for 1/12 and 3/12.

The frozen M3 decision was `MIXED_MODEL_BY_VARIANT_INTERACTION`. Coordinate-frame wording imposed a measurable burden, and explicit formulae produced deployment-specific effects. The control therefore reveals an interaction between semantic interpretation, geometric operator execution, and deployment behavior.

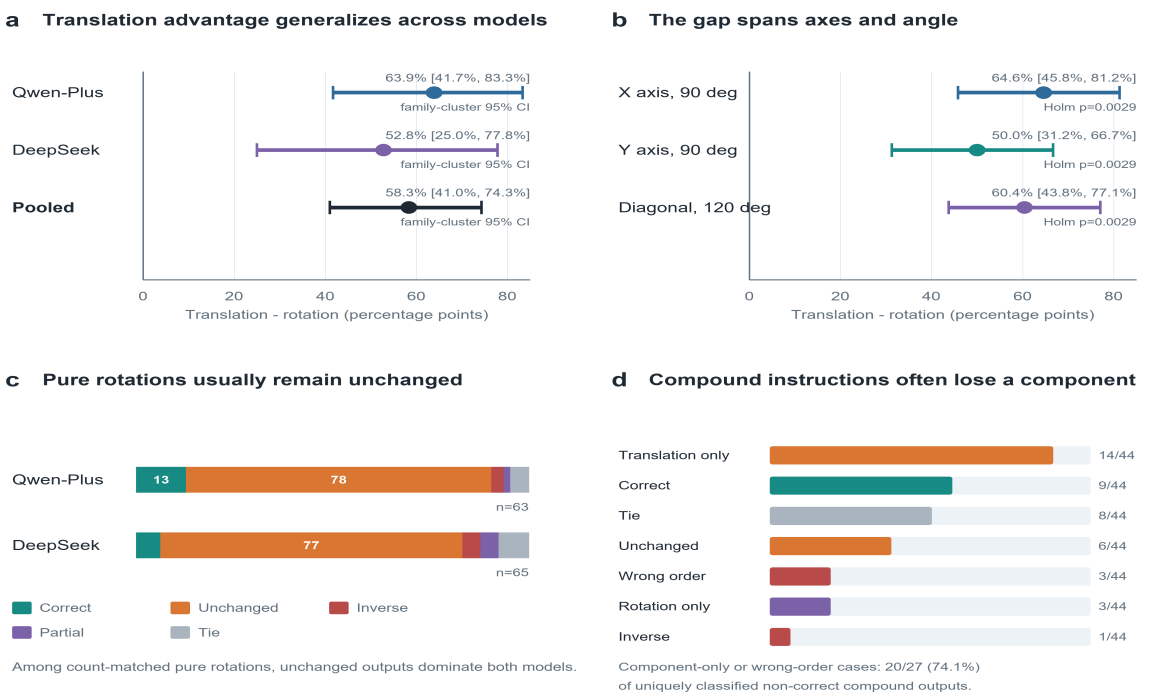


Prospective confirmation generalized beyond the original Z-axis rotation

M6 tested whether the earlier result was an artifact of the original Z-axis operation. In the prespecified intent-to-treat analysis, Qwen executed 18/24 translations (75.0%) and 8/72 pure rotations (11.1%), a difference of 0.639 (95% family-cluster bootstrap CI, 0.417-0.833). DeepSeek executed 14/24 translations (58.3%) and 4/72 rotations (5.6%), a difference of 0.528 (95% CI, 0.250-0.778). Pooled at the family level, 32/48 translations (66.7%) and 12/144 rotations (8.3%) were correct, giving the primary difference of 0.583 (95% CI, 0.410-0.743; Figure 4a; Table 3). The result met the predeclared `BROAD_ROTATION_GAP` rule: the pooled lower confidence bound exceeded zero and both deployment estimates were positive.

The gap appeared for each new pure rotation (Figure 4b; Table 4). Relative to the common fresh-translation rate, the pooled difference was 0.646 for X-axis 90-degree rotation (95% CI, 0.458-0.812), 0.500 for Y-axis 90-degree rotation (95% CI, 0.312-0.667), and 0.604 for diagonal-axis 120-degree rotation (95% CI, 0.438-0.771). The corresponding two-sided exact sign-flip tests all had Holm-adjusted $p = 0.0029$. Qwen's deployment-specific intervals excluded zero for all three rotations. DeepSeek's X-axis contrast remained below 0.05 after correction, while its Y-axis and diagonal contrasts had adjusted $p = 0.0625$ despite positive point estimates and intervals bounded above zero under the cluster bootstrap. This difference illustrates why the paper reports effect sizes, intervals, and deployment-specific results rather than treating one significance threshold as the result.

Active/passive representation accounted for little of the gap. In intent-to-treat M6 cells, active and passive translation correctness was identical within both deployments (9/12 for Qwen and 7/12 for DeepSeek). Pure-rotation active-passive differences were zero for DeepSeek and ranged from -0.083 to -0.167 for Qwen, with passive prompts slightly higher. The operation contrast held under both frame conventions.



Confirmatory errors preserved geometry or dropped the rotation component

The no-op-like pattern strengthened in M6 (Figure 4c; Table 5). Among count-matched pure-rotation outputs, Qwen produced 49/63 unchanged classifications (77.8%) and 2/63 inverse classifications (3.2%). DeepSeek produced 50/65 unchanged (76.9%) and 3/65 inverse (4.6%). Combined, 99/128 outputs (77.3%) were unchanged and 5/128 (3.9%) inverse. Correct rotation accounted for 12/128 (9.4%).

Compound transformations exposed a related compositional failure (Figure 4d). Among 44 count-matched outputs, 9 were correct (20.5%), 14 translation-only, six unchanged, three rotation-only, three wrong-order, one inverse, and eight ties. Restricting the denominator to uniquely classified non-correct outputs, 20/27 (74.1%) completed only one component or used the wrong order. Translation-only was the largest single error class. Thus, the same operation that was weak in isolation was often the component lost during composition.

Sensitivity analyses preserved the primary gap

The pooled intent-to-treat difference remained large across every frozen relation threshold (Table 6): 0.632 at epsilon=0, 0.632 at 0.001, 0.583 at 0.01, and 0.549 at 0.05. Conditioning on strict, count-matched outputs increased the primary pooled difference to 0.727 (95% CI, 0.518-0.886). Conditioning on strict objects without the count requirement yielded 0.693 (95% CI, 0.480-0.877).

The official C2CAD scorer produced a smaller but still substantial pooled intent-to-treat difference of 0.472 (95% CI, 0.299-0.646). Its deployment estimates were 0.528 (95% CI, 0.292-0.764) for Qwen and 0.417 (95% CI, 0.181-0.653) for DeepSeek. No official-scorer evaluation error occurred in M6. Across endpoint availability, JSON parsing, exact shape count, relation tie threshold, and scorer choice, the operation gap retained the same direction and practical consequence.

Discussion

Principal finding

FrameShift-CAD identifies a reproducible operation-specific weakness in structured text-to-CAD generation. The initial 18-family matrix established the direction across three model families, and the frozen M6 experiment confirmed a larger gap across two stable deployments, a fresh translation, and rotations spanning two principal axes and one diagonal axis. Exact translation succeeded in two thirds of confirmatory outputs; pure rotation succeeded in fewer than one in ten. The gap determines whether the requested coordinate operation is executed.

The error distribution is as important as the rate. If the models had mainly selected the opposite active/passive convention, inverse outputs would have dominated. They did not. More than three quarters of count-matched pure-rotation outputs were closest to the unchanged target, and inverse outputs were rare. Compound instructions sharpened the interpretation: translation was frequently preserved while rotation was omitted. The defensible output-level conclusion is under-execution of rotation, not universal confusion about the sign of a frame transform.

The prompt controls clarify a second source of difficulty. Terminology-only prompts were clearly difficult, so words such as *active*, *passive*, *object*, and *frame* impose semantic demands. Yet formula-only prompts retained positive translation-rotation gaps in both deployments and did not improve DeepSeek rotation. Explicit coordinate mappings are therefore useful but insufficient. In the generated geometry, the residual deficit is compatible with several sources, including planning, arithmetic, field binding, long-output consistency, and generation policy.

Relation to neighboring benchmarks

Existing work establishes coordinate-frame and workplane alignment as CAD failure sources and demonstrates deterministic symbolic repair (Zhang et al., 2026). FrameShift-CAD adds controlled quantification: the same design content is transported under matched operations, the translation-rotation contrast is estimated at the family level, new axes and an angle are prospectively confirmed, and failures are assigned to explicit counterfactuals.

FrameShift-CAD also differs from SGP-Bench. SGP-Bench perturbs the symbolic program and asks whether semantic answers remain invariant (Qiu et al., 2025). The present task holds a text-described design constant, asks the model to generate a transformed structured CAD output, and evaluates the generated coordinates. SpinBench provides strong adjacent evidence that rotation is difficult for vision-language spatial reasoning (Zhang et al., 2025), but its multiple-choice visual tasks do not require construction of complete CAD geometry. FoR-T2I shows that paired frame descriptions can expose large generation gaps (Gu et al., 2026), while FrameShift-CAD replaces image-layout judgment with exact three-dimensional transforms and machine-checkable geometric alternatives.

CADTestBench and CADEngBench make the most important methodological comparison. Both argue, in different ways, that appearance or single-reference similarity is inadequate: the former checks prompt requirements with executable predicates, and the latter evaluates parametric behavior, assemblies, and simulation (Mallis et al., 2026); (Singh et al., 2026). FrameShift-CAD follows the same move from static appearance to behavior, but its behavior is equivariant transport under a declared coordinate operation. The method can be layered onto richer B-rep and programmatic benchmarks rather than competing with them as a general replacement.

Implications for benchmark design

A single untransformed prompt cannot establish coordinate competence. A model may produce a plausible base object yet fail to move that object when every required mapping is stated explicitly. Benchmark suites should therefore include matched interventions and report operation families separately. Pooling translations and rotations into a generic “spatial” score would conceal the 58-point difference observed here.

Diagnostic counterfactuals should also be preserved. Correctness alone cannot distinguish sign reversal, no operation, partial application, component omission, and wrong order. Those categories lead to different engineering actions. A sign-reversal pattern suggests convention disambiguation. An unchanged pattern suggests verification that a requested operation entered the generation plan and reached every relevant field. A component-only pattern suggests explicit composition or intermediate-state checking.

Finally, protocol reliability must remain separate from geometric capability. The early non-streaming pilot mixed gateway timeouts with output-format violations. Streaming and JSON-object mode substantially improved transport, but M6 still contained parse and count failures. Intent-to-treat results appropriately capture endpoint reliability for a user-facing system, whereas conditional results estimate geometry among outputs that reached a specified protocol state. Reporting both avoids treating network failures as evidence of poor spatial reasoning while also avoiding an unrealistically clean denominator.

Implications for text-to-CAD systems

The unchanged and translation-only patterns suggest several testable system interventions. A generator could externalize a coordinate-operation plan before emitting shapes, represent the design as a scene graph with typed position and direction fields, and apply the final affine map deterministically after base-geometry construction. A verification loop could compare the emitted shape set against correct and unchanged counterfactuals and reject a no-op-like result. For compound operations, retaining an explicit homogeneous-transform chain would make order visible and mechanically checkable.

Execution-guided systems already show that CAD generation can benefit from deterministic feedback. CADLoop repairs near-miss programs by separating semantic intent from continuous geometric execution (Zhang et al., 2026), and RA-CAD learns to use post-execution critique during rewriting (Yan et al., 2026). FrameShift-CAD supplies a set of failure-specific signals that could condition such loops. Whether those signals improve generation is a future intervention study, not a result of the present evaluation.

Limitations

The study has six main limits. First, M6 confirms breadth over two deployments, not over the full model market. MiMo supplied a third-family directional replication only for the original operation, and its confidence interval crossed zero. The results should be attributed to the tested dated endpoints rather than generalized to all language models.

Second, every cell was sampled once at temperature zero. The experiment estimates deterministic endpoint behavior under a frozen prompt and access window. It does not estimate within-model sampling variance. Provider-side updates, routing changes, and unreported model revisions could affect future replication even when an API name is unchanged.

Third, all source cases are Level 1. This choice made exact intervention and oracle identification possible across multi-shape families, but it excludes higher-level topology, feature histories, local workplanes, manufacturing constraints, and realistic industrial assemblies. The next external-validity test should apply the same protocol to programmatic or B-rep tasks such as those represented in CADTestBench or CADEngBench.

Fourth, the relation oracle is intentionally narrow. It scores fields expected to change after canonicalization and does not measure every topological, semantic, or manufacturability defect. Its greedy fallback alignment may also be imperfect when identifiers are missing and repeated primitives are spatially ambiguous. Agreement with the official scorer and the stable sensitivity estimates reduce dependence on one metric but do not remove assumptions shared by reference-based geometry evaluation.

Fifth, the unchanged label is relative to the public untransformed target, not to the same deployment's independently sampled base output. It is therefore evidence of no-op-like geometry, not direct evidence that a model copied an earlier response. Likewise, the alternating-shape partial oracle represents only one form of incomplete transformation. The taxonomy is diagnostic but not exhaustive.

Sixth, M6 covers signed-permutation rotations and one composition. It does not exhaust SE(3), arbitrary angles, local coordinate frames, screw motions, long transform chains, or scale and reflection. The diagonal 120-degree condition broadens the tested angle and axis while retaining exact schema expressibility; it should not be read as evidence that every possible rotation has the same difficulty.

Conclusion

FrameShift-CAD turns coordinate operations into controlled, machine-checkable interventions for structured text-to-CAD generation. Across the tested Level-1 families and deployments, exact translations were executed far more often than rotations. The confirmatory gap persisted across X-axis, Y-axis, and diagonal-axis rotations, remained under intent-to-treat and scorer sensitivities, and was dominated by geometry closest to the unchanged target. When rotation was composed with translation, the rotation component was frequently lost.

The main contribution is a reproducible way to measure operation transport and distinguish the resulting errors. It creates a concrete path from benchmark score to system diagnosis: verify that the operation occurred, identify the failed component, and execute deterministic transforms outside unconstrained generation when exactness is required.

Data availability

The study uses the public C2CAD-Bench cases and derived local transformation targets. The reproducibility package is available at https://github.com/shushuyang231/frameshift-cad. It contains frozen request manifests, per-request provenance and hashes, scored rows, aggregate analysis files, figure/table transformations, and integrity reports. Provider credentials, secrets, and raw model-response bodies are excluded. C2CAD-derived files retain the upstream attribution and licence notice. A versioned archival DOI will be added before submission.

Code availability

Transformation generation, canonicalization, counterfactual scoring, bootstrap analysis, exact sign-flip tests, figure generation, and package validation are implemented in the public source repository at https://github.com/shushuyang231/frameshift-cad.

Ethics declaration

This study used public benchmark specifications and model API outputs. It involved no human participants, personal data, animals, or clinical materials. Institutional review-board approval and informed consent were therefore not applicable.

Author contributions

Shengyao Sun: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization, writing - original draft, writing - review and editing, and project administration. The author approved the final version of the manuscript.

Funding

The author received no specific funding for this study.

Declaration of competing interest

The author declares no competing interests.

Declaration of generative AI and AI-assisted technologies in the writing process

Generative AI tools were used to assist experimental software development, literature discovery, and preparation of the manuscript draft. The authors inspected the source materials, controlled all external model calls, verified the numerical analyses against deterministic local artifacts, revised the text, and remain responsible for the accuracy, interpretation, and

final submitted content.

References

C2CAD-Bench Research Team. (2026). *C2CAD-Bench: A large-scale benchmark for 3D spatial reasoning in CAD* [Benchmark website]. Retrieved September 2, 2026, from https://c2cad-bench.github.io/

Chen, T. Y., Cheung, S. C., & Yiu, S. M. (1998). *Metamorphic testing: A new approach for generating next test cases* (Technical Report HKUST-CS98-01). Hong Kong University of Science and Technology. https://www.cse.ust.hk/faculty/scc/publ/CS98-01-metamorphictesting.pdf

Gu, Z., Li, R., Huang, Y., Zhang, Y., Hao, X., Niu, J., Hu, J., & Zhang, Z. (2026). Can text-to-image models draw from the right frame of reference? *arXiv*. https://doi.org/10.48550/arXiv.2608.03357

Guan, Y., Wang, X., Ming, X., Zhang, J., Xu, D., & Yu, Q. (2025). CAD-Coder: Text-to-CAD generation with chain-of-thought and geometric reward. *arXiv*. https://doi.org/10.48550/arXiv.2505.19713

Khan, M. S., Sinha, S., Sheikh, T. U., Stricker, D., Ali, S. A., & Afzal, M. Z. (2024). Text2CAD: Generating sequential CAD models from beginner-to-expert level text prompts. *arXiv*. https://doi.org/10.48550/arXiv.2409.17106

Li, J., Ma, W., Li, X., Lou, Y., Zhou, G., & Zhou, X. (2025). CAD-Llama: Leveraging large language models for computer-aided design parametric 3D model generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 18563-18573). https://openaccess.thecvf.com/content/CVPR2025/html/Li_CAD-Llama_Leveraging_Large_Language_Models_for_Computer-Aided_Design_Parametric_3D_CVPR_2025_paper.html

Mallis, D., Wang, M., Karadeniz, A. S., Ricci, E., Kacem, A., & Aouada, D. (2026). Text-to-CAD evaluation with CADTests. *arXiv*. https://doi.org/10.48550/arXiv.2605.07807

Pyatov, V., Bobrovskikh, G., Galochkin, S., Boldyrev, N., Voynov, O., Filippov, A., Ferrer, G., Wonka, P., & Burnaev, E. (2026). CADFS: A big CAD program dataset and framework for computer-aided design with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10176-10186). https://openaccess.thecvf.com/content/CVPR2026/html/Pyatov_CADFS_A_Big_CAD_Program_Dataset_and_Framework_for_Computer-Aided_CVPR_2026_paper.html

Qiu, Z., Liu, W., Feng, H., Liu, Z., Xiao, T. Z., Collins, K. M., Tenenbaum, J. B., Weller, A., Black, M. J., & Schölkopf, B. (2025). Can large language models understand symbolic graphics programs? *International Conference on Learning Representations*. https://openreview.net/forum?id=Yk87CwhBDx

Singh, H., Dubey, A., & Amador Herrera, J. A. (2026). CADEngBench: It looks like CAD, but does it work? Evaluating parametric design, assembly reasoning, and physics simulation. *arXiv*. https://doi.org/10.48550/arXiv.2608.09296

Wang, L., Meng, H., Xiang, Z., Liu, J., Zhou, P., Chen, L., & Tang, Y. (2026). Text2CAD-Bench: A benchmark for LLM-based text-to-parametric CAD generation. *arXiv*. https://doi.org/10.48550/arXiv.2605.18430

Wang, S., Chen, C., Le, X., Xu, Q., Xu, L., Zhang, Y., & Yang, J. (2024). CAD-GPT: Synthesising CAD construction sequence with spatial reasoning-enhanced multimodal LLMs. *arXiv*. https://doi.org/10.48550/arXiv.2412.19663

Yan, S., He, C., Peng, X., & Hu, P. (2026). RA-CAD: Learning post-execution critique for state-aware text-to-CAD generation. *arXiv*. https://doi.org/10.48550/arXiv.2608.05714

Zhang, H., Chen, M., & Liu, K. (2026). CADLoop: An equivariant-aware skill-grounded loop for CAD data curation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 3092-3100). https://openaccess.thecvf.com/content/CVPR2026W/NeXD/html/Zhang_CADLoop_An_Equivariant-Aware_Skill-Grounded_Loop_for_CAD_Data_Curation_CVPRW_2026_paper.html

Zhang, Y., Corcodel, R., Hori, C., Cherian, A., & Zhao, D. (2025). SpinBench: Perspective and rotation as a lens on spatial reasoning in VLMs. *arXiv*. https://doi.org/10.48550/arXiv.2509.25390

Zheng, Z., Zhou, Z., Xu, Y., Ren, D., & Chen, T. Y. (2026). Bidirectional empowerment of metamorphic testing and large language models: A systematic survey. *arXiv*. https://doi.org/10.48550/arXiv.2605.13898

Appendix: compact result tables

Table A1. Initial 18-family operation gap.

model	translation correct	translation evaluable	rotation correct	rotation evaluable	translation minus rotation	ci low	ci high
qwen-plus	22	33	8	33	0.4444	0.1944	0.6667
deepseek-v4-pro	23	29	10	34	0.5	0.2647	0.7353
mimo-v2.5-pro	9	20	4	26	0.2727	-0.0909	0.6364
combined	54	82	22	93	0.4389	0.2287	0.6333

Table A2. Prompt-control correctness.

model	variant	translation correct	translation evaluable	rotation correct	rotation evaluable	translation minus rotation
qwen-plus	term_plus_formula	9	11	3	11	0.5833
qwen-plus	formula_only	9	11	5	12	0.4167
qwen-plus	terminology_only	6	11	0	12	0.5
deepseek-v4-pro	term_plus_formula	8	11	5	12	0.3333
deepseek-v4-pro	formula_only	6	12	3	11	0.25
deepseek-v4-pro	terminology_only	7	11	1	12	0.5833

Table A3. M6 primary estimates.

model	analysis	translation correct	translation n	rotation correct	rotation n	translation minus rotation	ci low	ci high
qwen-plus	itt	18	24	8	72	0.638889	0.416667	0.833333
qwen-plus	conditional_count_match	18	22	8	63	0.691198	0.465909	0.865519
qwen-plus	strict_only	19	24	9	69	0.661232	0.463235	0.831522
deepseek-v4-pro	itt	14	24	4	72	0.527778	0.25	0.777778
deepseek-v4-pro	conditional_count_match	14	17	4	65	0.761991	0.491429	0.971014
deepseek-v4-pro	strict_only	14	18	4	71	0.72144	0.442857	0.971429
pooled	itt	32	48	12	144	0.583333	0.409722	0.743056
pooled	conditional_count_match	32	39	12	128	0.726763	0.517912	0.885611
pooled	strict_only	33	42	13	140	0.692857	0.479553	0.87739

Table A4. Rotation-specific M6 contrasts.

model	rotation	translation minus rotation	ci low	ci high	holm adjusted p
qwen-plus	rx90	0.708333	0.5	0.916667	0.005859
qwen-plus	ry90	0.5	0.25	0.708333	0.008789
qwen-plus	diag120	0.708333	0.458333	0.916667	0.005859
deepseek-v4-pro	rx90	0.583333	0.333333	0.833333	0.046875
deepseek-v4-pro	ry90	0.5	0.25	0.75	0.0625
deepseek-v4-pro	diag120	0.5	0.25	0.75	0.0625
pooled	rx90	0.645833	0.458333	0.8125	0.00293
pooled	ry90	0.5	0.3125	0.666667	0.00293
pooled	diag120	0.604167	0.4375	0.770833	0.00293