Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

From Tell-to-Design to Healthcare Test-Fit Constraint Checking: An LLM-Compatible Requirements-to-Constraints Framework

Justin Sun[1], Xiaoming Xiao[1], Huichao Dong[2]
[1]*Applied Analytics, Columbia University, NY, USA*
[1]*College of Civil Engineering, Hunan University, Changsha 410082, China*
[2]*Department of Architecture, University of Pennsylvania, Philadelphia, PA, USA*
justin.sun0812@gmail.com

DOI: 10.69987/JACS.2023.31105

Keywords

healthcare architecture; clinical user group; early-stage test-fit; constraint checking; behavioral health crisis unit; emergency department planning; natural language processing; large language models; room data sheets; adjacency constraints

Abstract

Language-guided floor-plan generation is attractive for early design, but hospital and behavioral health planning should not be treated as direct automatic plan generation. Early healthcare test-fits depend on jurisdictional review, owner standards, infection prevention, behavioral health safety, staff workflow, and clinical risk judgments that exceed a simple room-list prediction task. This paper reframes the problem as requirements-to-constraints translation followed by explicit constraint checking for early-stage test-fit design. We compiled HPEV-401, a source-grounded corpus of 401 emergency and behavioral health planning statements from four public healthcare planning sources: the International Health Facility Guidelines Emergency Unit functional planning unit, a room-data-sheet compendium, the U.S. Department of Veterans Affairs Emergency Department space planning criteria, and the Facility Guidelines Institute behavioral health crisis unit white paper. Each record preserves source and page metadata and is mapped to room-program, adjacency, and risk labels. Three translators were evaluated: a keyword baseline, a TF-IDF one-vs-rest logistic regression model, and a hybrid structured translator that combines lexical evidence, supervised multi-label prediction, and schema-level normalization. A graph-based checker then evaluated 324 candidate test-fit graphs created from the held-out split under four perturbation levels. The hybrid translator achieved held-out micro-F1 of 0.874 for room-program extraction, 0.744 for adjacency extraction, and 0.762 for risk extraction. With gold constraints, the checker detected all injected violations across nonzero severities. With hybrid-predicted constraints, the checker reached 0.956 recall at the highest perturbation severity. The results support a cautious design-assistance position: language models can structure clinical intent and route it to externally maintained profiles, while deterministic checkers expose conflicts for professional review rather than asserting code compliance.

Introduction

Natural language has become a practical interface for design computing because clinical leaders, facility managers, planners, and architects often express early requirements in narrative form rather than as complete drawings. In a conventional residential or office-planning setting, a language-guided model can be asked to generate a floor plan from room names and adjacency preferences. In healthcare, that framing is too narrow. A behavioral health intake zone, emergency department fast-track pod, decontamination entry, isolation pod, or forensic examination pathway must be evaluated against

space, risk, infection-control, visibility, acoustic, security, and workflow considerations. Healthcare planning sources such as iHFG emergency planning guidance, room data sheets, VA emergency department criteria, and FGI behavioral health crisis guidance present these requirements as a mixture of room programs, relationships, environmental requirements, and risk controls [1]-[4].

This distinction is especially important for work inspired by language-guided design datasets. Residential floor-plan corpora and text-to-house studies show that natural language can guide room placement, connectivity, and broad spatial semantics [21]-[23]. Healthcare test-fit planning uses similar language at the interface, but the decision target is different. The useful output is not a finished plan image. It is a structured set of planning constraints that can be checked, challenged, and revised by the project team. A single clinical sentence can imply a room, an adjacency, a separation rule, a risk-control assumption, and a profile check. Treating that sentence as a drawing prompt hides those decisions; translating it into constraints exposes them.

The safer research question is therefore not whether an LLM [25] can automatically generate a compliant hospital floor plan. The question is whether a language model or language-oriented translator can convert planning statements into a structured constraint set that a deterministic checker and human design team can review. This reframing follows the same motivation as evidence-based healthcare design: layout and environmental decisions influence safety, privacy, staff effectiveness, infection risk, and patient dignity [12], [13]. It also aligns with the design-process reality that clinical groups often state preferences such as line-of-sight supervision, separation of public and ambulance arrivals, avoidance of elopement paths, and protection of acoustic privacy before the project team has a fixed plan.

The paper contributes an empirical framework for this requirements-to-constraints problem. First, it defines a schema for room types, adjacency rules, and risk rules that can be checked against graph-based test-fit representations. Second, it treats healthcare planning criteria and clinical-user preferences as external profiles rather than as hidden model knowledge. Third, it evaluates the translation and checking pipeline on a source-grounded corpus compiled from public healthcare planning documents, rather than on a template-only requirement set. Figure 1 shows the resulting workflow.

The clinical-user-group framing is important because early conversations are heterogeneous. An emergency physician may describe the same planning issue as patient throughput, a nurse manager may describe it as observation from the team station, a security lead may describe it as controlled egress, and an infection-prevention reviewer may describe it as clean/dirty separation. A direct text-to-plan generator would collapse these voices into a single drawing proposal. The proposed pipeline keeps them as auditable constraints, so a design team can see which statement produced a room, which statement produced an adjacency, and which external profile introduced a project-level check.

This work deliberately avoids two claims. It does not claim that a model trained on general floor-plan descriptions understands healthcare codes. It also does not claim that a candidate test-fit accepted by the checker is code-compliant. The checker evaluates only the explicit constraints supplied by the schema and profiles. Final decisions remain with licensed professionals, infection-prevention reviewers, life-safety reviewers, state health departments, and authorities having jurisdiction. Table 1 summarizes this scope boundary.

Table 1. Scope correction from floor-plan generation to requirement translation.

Risky claim avoided	Operational replacement used in this paper	Reason
Automatic compliant hospital plan generation	Natural-language requirement translation plus deterministic constraint checking	Regulatory adoption and project approval are external conditions, not model memory.

Risky claim avoided	Operational replacement used in this paper	Reason
Room list equals clinical program	Room program plus adjacency, separation, visibility, access, and risk rules	Emergency and behavioral health planning require workflow and risk logic.
LLM decides compliance	External profiles provide project-specific constraints for professional review	Profiles are inspectable and replaceable by jurisdictional or owner criteria.
Generated plan is final design	Checker produces a violation report for early test-fit iteration	Human review remains responsible for clinical, regulatory, and constructability decisions.

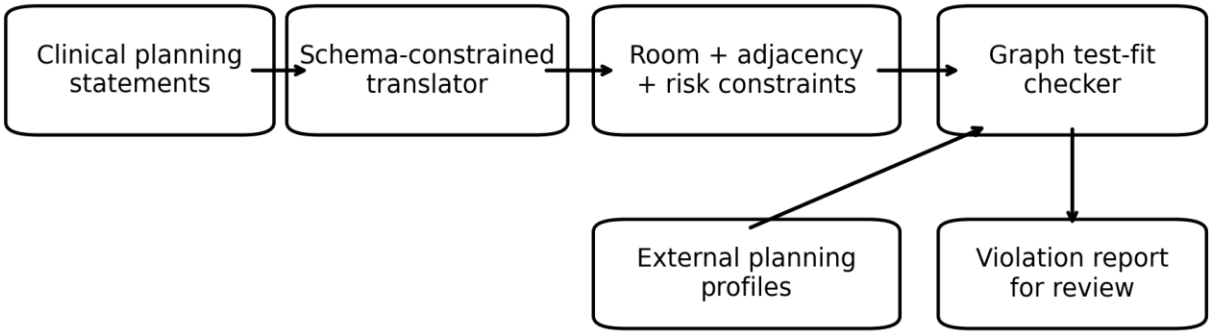

Figure 1. Requirements-to-constraints framework for healthcare test-fit planning.

Method

The method has three layers. The first layer translates free-text planning statements into a normalized output schema. The second layer merges the translated constraints with an external profile selected for the project context. The third layer checks a candidate test-fit graph and returns explicit violations. The graph representation uses nodes for rooms, undirected edges for required near-adjacency, avoided edges for separation constraints, and Boolean flags for project-level conditions such as controlled access, staff visual control, acoustic privacy, infection control, anti-ligature assumptions, and clean/dirty separation.

The graph abstraction was selected because it matches early test-fit practice. At the first feasibility stage, the team often knows whether observation should be near staff work, whether a mental health or secure holding room should avoid public waiting, and whether patient and staff/service paths should be separated, but it may not yet know exact wall positions, door swings, or detailed dimensions. A graph can therefore encode decisions that are stable enough for early review while postponing geometry that belongs to schematic design.

The empirical implementation used deterministic and supervised translators rather than a closed external LLM API. This makes the evaluation repeatable and keeps the translation layer

inspectable. The architecture is LLM-compatible because an LLM can replace the text translator when it emits the same JSON schema. The experiments therefore evaluate the structured-output and checking layer required before any LLM output is used in a healthcare planning workflow. Transformer-based language models supply the background motivation for schema-guided text processing [17]-[20], while graph-constrained layout work motivates explicit spatial relations rather than image-only outputs [23].

Source-grounded corpus

HPEV-401 contains 401 planning statements compiled from four public healthcare planning sources. The sources cover emergency unit functional planning, room data sheets, VA emergency department space planning criteria, and behavioral health crisis unit guidance [1]-[4]. Each record is a sentence, table-row statement, or room-data-sheet requirement with source and page metadata. Records were retained when they expressed at least two observable planning concepts in the output schema. Boilerplate, reference lists, and fragments without a room, adjacency, or risk signal were excluded.

The split was stratified by source with random seed 42. Table 2 lists the resulting 241 training records, 79 development records, and 81 held-out test records. Figure 2 visualizes the same split. The development split was used only for threshold tuning of the TF-IDF logistic regression model. The held-out split contains wording from all four source families, including definitions, room requirements, functional relationships, and behavioral health planning statements.

Table 2. HPEV-401 source-grounded corpus split by source.

Source	Content used	Train	Dev	Test	Total
iHFG Emergency	Emergency unit functional planning and schedule of accommodation	65	21	22	108
FGI BHCU	Behavioral health crisis unit planning, spaces, security, and workflow	65	22	21	108
VA ED	VA ED space planning criteria, room criteria, and functional relationships	65	21	22	108
Room Data Sheets	Room-level data sheets covering room details, services, and special requirements	46	15	16	77

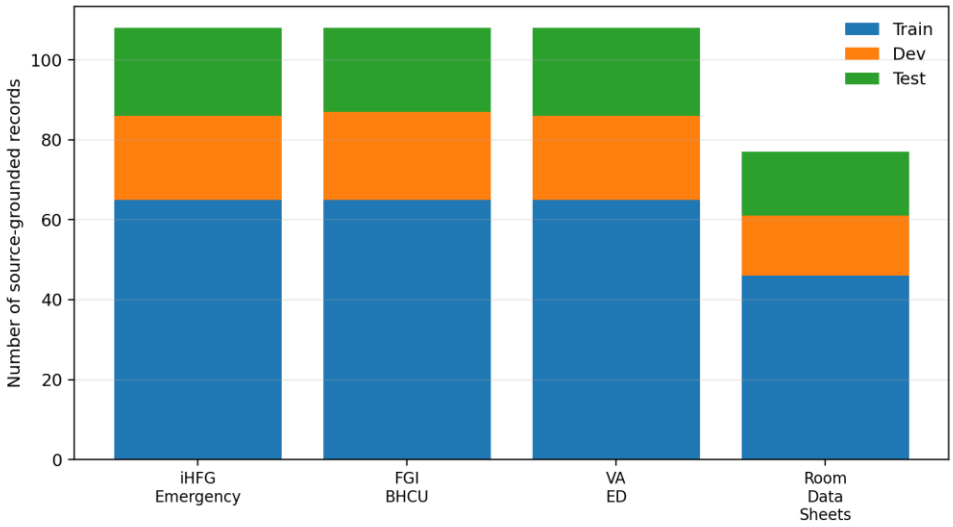

Figure 2. HPEV-401 split by source document family.

The corpus was mapped to three label families: room program, adjacency/separation, and risk/operational rules. The mapping was intentionally limited to information expressible in early test-fit review. For example, a statement that a decontamination area should be directly accessible from ambulance bays maps to a decontamination room label, a decontamination-near-ambulance adjacency, and an infection-control risk. A statement that a behavioral health crisis unit is controlled and visible from a nurse station maps to behavioral health, staff work, controlled-access, and visibility constraints. Table 3 summarizes the output schema, and Figure 3 shows the highest-support labels in the corpus.

Table 3. Output schema evaluated in the experiments.

Category	Number of labels	Examples
Room program	26	triage; staff_work_area; secure_holding; isolation; decontamination; consult_exam
Adjacency and separation	11	triage_near_waiting; public_ambulance_separated; clean_soiled_separated; bhcu_near_ed
Risk and operational rules	14	controlled_access; high_visibility; acoustic_privacy; infection_control; clean_dirty_separation

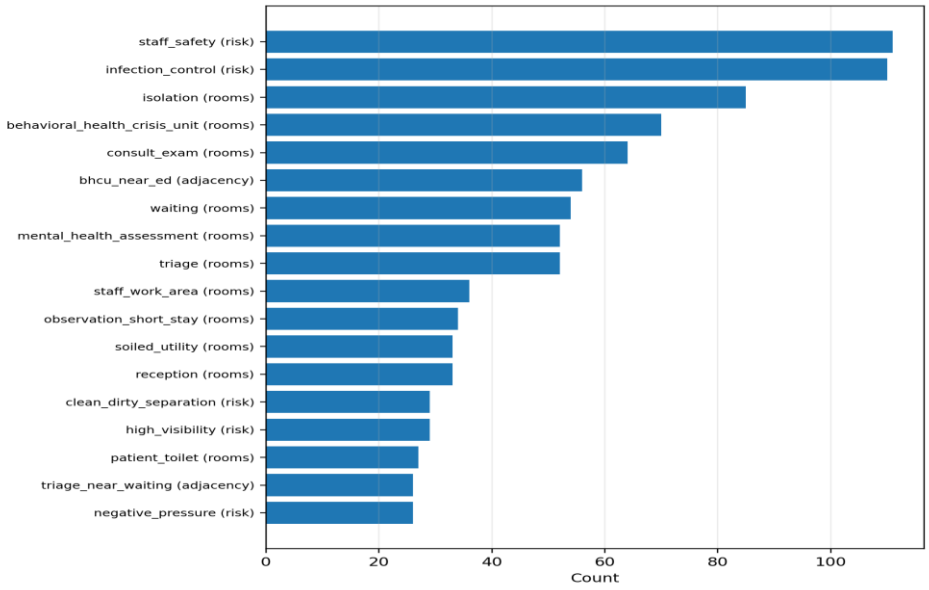

Figure 3. Top room, adjacency, and risk labels in HPEV-401.

External profiles

Profile handling is deliberately separated from text interpretation. The translator reads a planning statement and predicts what that statement appears to request. The profile layer states what must also be reviewed because of project context. In the experiment, each source family carries a profile that adds review flags to the checker. These profiles demonstrate external constraint injection; they do not reproduce full regulatory language or replace professional interpretation. Table 4 lists the profiles used by the checker.

Table 4. External constraint profiles used by the checker.

Profile	Purpose	External checks
iHFG_Emergency	Emergency unit planning profile for entry separation, infection prevention, and treatment-zone review	profile_ihfg_emergency_unit_review; risk_assessment_required; infection_prevention_review
Room_Data_Sheets	Room-level profile for minimum room information, services, and special requirements	profile_room_data_sheet_review; room_data_sheet_services_review
VA_ED	Emergency department space planning profile with security, workload, and room-criteria checks	profile_va_ed_space_planning_review; security_screening_review; ed_space_planning_review
FGI_BHCU	Behavioral health crisis unit profile emphasizing safety risk assessment, staff visibility, and controlled access	profile_fgi_bhcu_review; safety_risk_assessment_required; staff_visibility_review

Translators

Three translators were compared. The Keyword baseline searches for a deliberately small set of hand-written lexical cues. TFIDF-LR uses word and bigram TF-IDF features and one-vs-rest logistic regression for each output category. The Hybrid translator unions lexical and supervised predictions and applies schema-level normalization. For example, a decontamination room implies an infection-control review and an ambulance-side decontamination adjacency; clean and soiled utility labels imply clean/dirty separation; and behavioral health crisis language adds staff-safety review. These normalizations are explicit schema rules rather than hidden prompt decisions. Table 5 summarizes the translator configurations.

Table 5. Translator configurations.

Model	Input features	Training data	Output handling
Keyword	Lowercase phrase matching	No fitted parameters	Returns labels whose cue phrases appear in the statement
TFIDF-LR	1- and 2-gram TF-IDF, max 6000 features	241 training records; dev threshold tuning	One-vs-rest logistic regression per label category
Hybrid	Keyword evidence plus TFIDF-LR predictions	Same fitted model as TFIDF-LR	Union plus schema normalization and risk-rule closure

Graph checker and candidate plans

The checker evaluates candidate test-fit graphs. A satisfying graph is created from each held-out test record by adding required room nodes, required near-adjacency edges, absent forbidden edges, and required profile flags. Four perturbation severities are then produced: severity 0 leaves the plan clean, severity 1 injects up to two violations, severity 2 injects up to four violations, and severity 3 injects up to six violations. Perturbations are generated by deleting required adjacency edges, adding forbidden adjacency edges, or turning off required flags. The experiment therefore evaluates 81 test records under four severities, yielding 324 candidate plans. For each plan, the checker is evaluated once with gold constraints and once with predicted constraints from each translator.

The checker itself is intentionally simple. A required-edge rule fails when the corresponding room-pair edge is absent. An avoid-edge rule fails when a forbidden room-pair edge is present. A flag rule fails when a required Boolean property is false. External profile checks are represented as required flags so that the same mechanism handles clinical-user preferences, source profiles, and project-level reviews. This simplicity is useful for design review: a false positive is a named edge or flag that can be inspected, suppressed, or converted into a project-specific requirement.

Metrics are micro-precision, micro-recall, micro-F1, macro-F1, exact match, and label accuracy for extraction; precision, recall, F1, true positives, false positives, and false negatives for plan checking. Micro-F1 emphasizes frequent labels, while macro-F1 exposes weak performance on lower-support but clinically meaningful concepts. Exact match is intentionally strict because an omitted risk or separation rule can prevent a complete review.

Results and Discussion

Table 6 and Figure 4 show the extraction comparison on the held-out test split. The keyword baseline performs well when room names appear in conventional wording, but it misses many adjacency

and risk paraphrases. TFIDF-LR improves recall for adjacency and risk labels. The Hybrid translator gives the best overall result, with micro-F1 of 0.874 for room-program extraction, 0.744 for adjacency extraction, and 0.762 for risk extraction. The gain is clearest for adjacency and risk extraction because schema normalization recovers implied relationships such as clean/dirty separation from clean and soiled utility language and infection-control review from decontamination or isolation language.

Table 6. Requirement extraction results on the held-out test split.

Category	Model	Micro-P	Micro-R	Micro-F1	Macro-F1	Exact	Label acc.
rooms	Keyword	1.000	0.709	0.829	0.561	0.556	0.979
rooms	TFIDF-LR	0.849	0.709	0.773	0.656	0.444	0.970
rooms	Hybrid	0.874	0.874	0.874	0.757	0.580	0.982
adjacency	Keyword	1.000	0.286	0.444	0.332	0.716	0.972
adjacency	TFIDF-LR	0.684	0.743	0.712	0.582	0.778	0.976
adjacency	Hybrid	0.674	0.829	0.744	0.687	0.802	0.978
risk	Keyword	1.000	0.082	0.152	0.165	0.247	0.931
risk	TFIDF-LR	0.776	0.694	0.733	0.597	0.568	0.962
risk	Hybrid	0.719	0.812	0.762	0.646	0.556	0.962

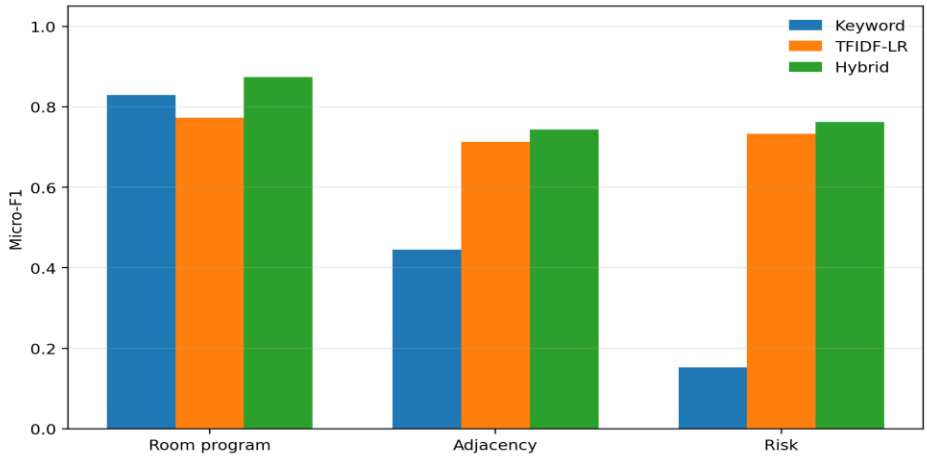

Figure 4. Micro-F1 comparison by model and output category.

The label-level results in Table 7 and Figure 5 show that high-support clinical concepts are reliably recovered, while some lower-support relationship labels remain difficult. Isolation, behavioral health crisis unit, triage, reception, and waiting labels are strong because the source documents name those spaces directly. Relationship labels are more sensitive to phrasing. For example, bhcu_near_ed appears in several forms, including "readily accessible to the emergency department" and "co-located with an emergency department," which are semantically close but lexically different.

Table 7. Selected Hybrid label-level test metrics.

Category	Label	Support	Precision	Recall	F1
adjacency	bhcu_near_ed	10	0.643	0.900	0.750
adjacency	triage_near_waiting	6	0.857	1.000	0.923
adjacency	patient_flow_separated	5	1.000	0.400	0.571
risk	infection_control	28	0.917	0.786	0.846
risk	staff_safety	22	0.880	1.000	0.936
risk	controlled_access	7	1.000	0.571	0.727
rooms	isolation	19	1.000	1.000	1.000
rooms	behavioral_health_crisis_unit	14	0.933	1.000	0.966
rooms	waiting	13	0.929	1.000	0.963
rooms	consult_exam	13	1.000	0.769	0.870
rooms	triage	11	0.917	1.000	0.957
rooms	reception	10	1.000	1.000	1.000

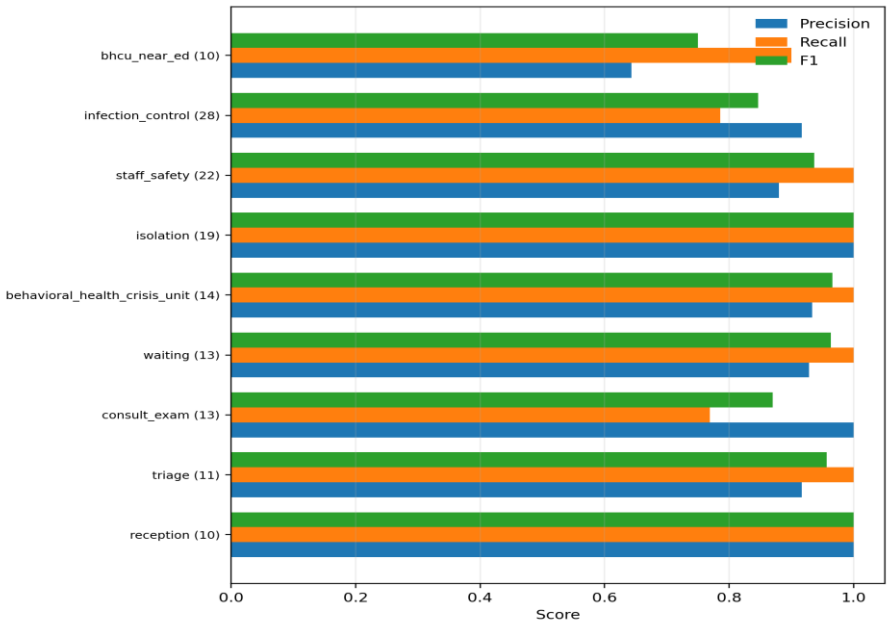

Figure 5. Hybrid label-level precision, recall, and F1 for selected high-support labels.

The results also clarify why room extraction alone is insufficient. A room label such as staff_work_area becomes useful only when paired with a visibility or observation rule. Similarly, isolation and decontamination rooms require infection-control and pressure/flow assumptions, and behavioral health spaces require controlled access, staff safety, and elopement review. The schema therefore evaluates rooms, adjacencies, and risks as separate but related outputs.

Plan-checking performance

The plan-checking experiment separates two questions. First, if a project team supplies correct constraints, does the checker find the injected plan violations? Second, if the checker receives automatically translated constraints, how much review value remains? Table 8 and Figure 6 report nonzero severity results. With gold constraints, the checker detected every injected violation. Using predicted constraints, the Hybrid checker achieved the highest recall at every nonzero severity. At severity 3, it detected 325 of 340 injected violations, reaching 0.956 recall and 0.917 F1.

Table 8. Constraint-checker performance on perturbed candidate plans.

Constraint source	Severity	Precision	Recall	F1	TP	FP	FN
Gold constraints	1	1.000	1.000	1.000	162	0	0
Gold constraints	2	1.000	1.000	1.000	294	0	0
Gold constraints	3	1.000	1.000	1.000	340	0	0
Keyword	1	1.000	0.778	0.875	126	0	36
Keyword	2	1.000	0.779	0.876	229	0	65
Keyword	3	1.000	0.785	0.880	267	0	73
TFIDF-LR	1	0.812	0.907	0.857	147	34	15
TFIDF-LR	2	0.887	0.912	0.899	268	34	26
TFIDF-LR	3	0.901	0.915	0.908	311	34	29
Hybrid	1	0.780	0.963	0.862	156	44	6
Hybrid	2	0.865	0.956	0.908	281	44	13
Hybrid	3	0.881	0.956	0.917	325	44	15

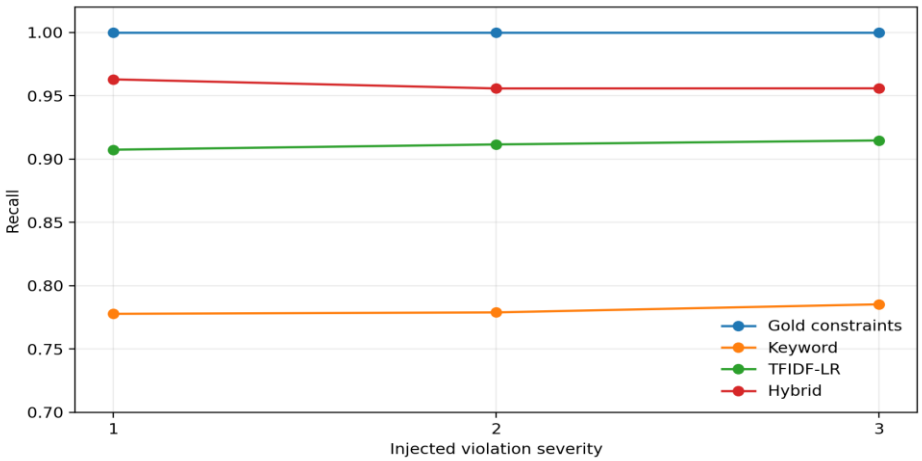

Figure 6. Constraint-checker recall across injected violation severities.

The severity trend matters. At low severity, a small number of missed or extra predicted constraints has a large effect on precision and recall. At higher severity, recall becomes a clearer measure of whether the translation layer preserves the important design intent. The Hybrid model produces more warnings than the keyword baseline, but those warnings are named and inspectable. In an early test-fit meeting, an extra warning about observation or controlled access is less damaging than a missed elopement, infection-control, or clean/dirty separation issue.

Clean-plan false alarms are summarized in Table 9, and the predicted-checker error counts are shown in Figure 7. A clean-plan warning is not treated as an automatic failure; it is a prompt for review. If the checker flags a clean graph because the translator inferred a plausible constraint not present in the source-mapped label set, the design team can accept it, suppress it, or add it to the project profile. This is appropriate for decision support, but it would not be appropriate for automated approval.

Table 9. Clean-plan false alarms at severity 0.

Constraint source	TP	FP	FN	Interpretation
Gold constraints	0	0	0	Clean candidate graphs; FP count is review burden.
Keyword	0	0	0	Clean candidate graphs; FP count is review burden.
TFIDF-LR	0	34	0	Clean candidate graphs; FP count is review burden.
Hybrid	0	44	0	Clean candidate graphs; FP count is review burden.

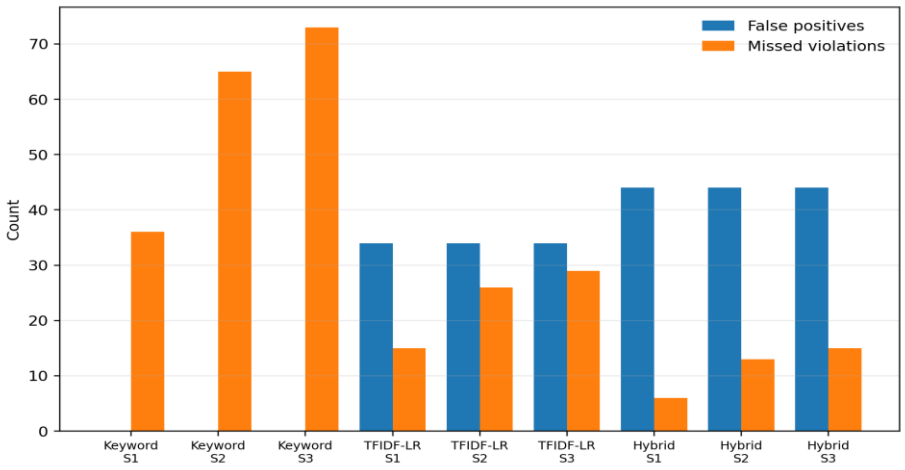

Figure 7. False positives and missed violations for predicted-constraint checking.

Ablation study

The ablation study in Table 10 and Figure 8 confirms that combining supervised evidence with schema normalization improves mean extraction performance over TFIDF-LR alone. The difference between Hybrid and Hybrid_no_ontology is modest because the source statements are often explicit, but normalization still improves mean micro-F1 from 0.787 to 0.793 and is useful for downstream checking. The keyword baseline remains useful as a precise signal for named rooms, but it is not sufficient for adjacency and risk extraction.

Table 10. Ablation results for the translation layer.

Variant	Room F1	Adjacency F1	Risk F1	Mean F1	Joint exact
Hybrid	0.874	0.744	0.762	0.793	0.370
Hybrid_no_ontology	0.874	0.730	0.756	0.787	0.370
TFIDF_LR_only	0.773	0.712	0.733	0.739	0.346
Keyword_only	0.829	0.444	0.152	0.475	0.074

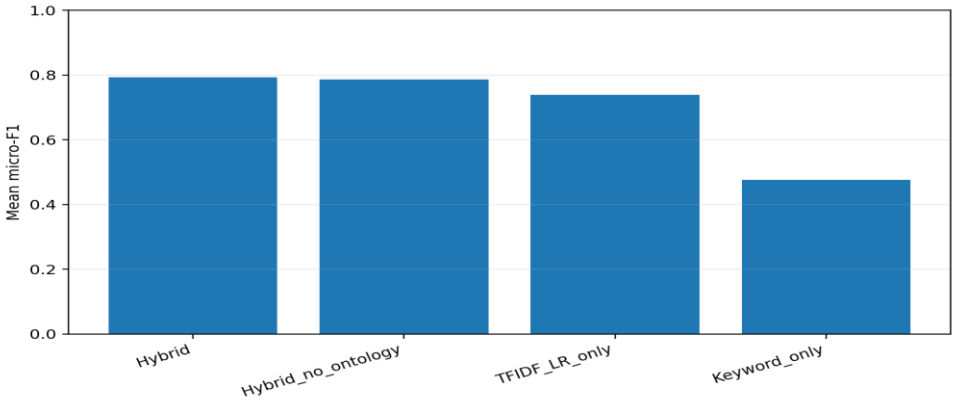

Figure 8. Ablation of translation components.

Analysis assets

Table 11 summarizes the assets used for the experiments. The dataset is compact enough to inspect record by record, while still covering a broad set of emergency, behavioral health, infection-control, security, and room-data-sheet statements. The candidate graphs are deliberately simple, because the purpose is to test translation-to-checking behavior before connecting the checker to CAD or BIM geometry.

Table 11. Analysis assets used in the study.

Asset	Value
Dataset	HPEV-401 source-grounded records in CSV and JSONL form
Splits	241 train, 79 development, 81 test
Source families	iHFG Emergency Unit; room data sheets; VA ED criteria; FGI BHCU white paper
Candidate plans	324 graph test-fit plans under four perturbation severities
Random seed	42
Runtime	5.92 seconds on the analysis environment
Scripts	experiment_build.py; make_revised_figures.py
Outputs	CSV metrics and regenerated PNG figures

A practical reading of the results is that the translator should be tuned for recall and the checker should be tuned for transparent review. The Hybrid model produces more clean-plan warnings than the keyword baseline, but those warnings are explicit edges or flags. In an early test-fit meeting, the cost of reviewing an extra visibility or access item is usually lower than the cost of missing a safety-related separation or infection-control condition. The workflow therefore favors surfacing potentially relevant constraints, then allowing the design team to accept, reject, or revise each item with reference to the selected profile and stakeholder statement.

External profiles are the mechanism that prevents the translator from becoming a hidden code consultant. A profile can be written, reviewed, versioned, and approved by the project team outside the model. During a project, the profile would be selected according to facility type, state adoption, owner standard, and service-line scope. If a state amendment or owner standard changes, the profile changes; the translator does not need to be retrained. This separation also supports copyright-sensitive implementation because project teams can store authorized interpretations in their own systems instead of requiring a public model to reproduce protected clauses.

The design-computing implication is that spatial generation and constraint checking should remain separate. Text-to-image and graph-constrained floor-plan models have advanced rapidly for residential layouts [21]-[23], and language-to-house work shows the appeal of natural language as a design interface [22]. Healthcare test-fit planning adds regulatory and clinical risk layers that require auditable constraint representation. By separating language interpretation from profile checking, the proposed method lets a project team update the profile when a state amendment, owner standard, or service-line preference changes without retraining the translator.

Limitations

The first limitation is corpus origin. HPEV-401 is grounded in public healthcare planning documents, not in confidential project meeting minutes. It improves external grounding compared with a purely synthetic benchmark, but it still does not capture the full ambiguity, disagreement, and incompleteness of real clinical user-group workshops. Future work should evaluate the same schema on de-identified project narratives, owner standards, room-data-sheet comments, and design-review meeting notes.

The second limitation is the translator. The evaluated implementation uses keyword rules, TF-IDF logistic regression, and deterministic normalization rather than a proprietary LLM. This choice keeps the structured-output layer transparent. In deployment, an LLM can be used as the front-end translator if it is constrained to emit the same schema and if its outputs are checked against the same external profiles. The reported results therefore validate the translation-and-checking framework, not the superiority of any specific LLM provider.

The third limitation is the constraint checker. The checker operates on simplified graph test-fits, not on BIM models, CAD drawings, door schedules, line-of-sight geometry, fire/life-safety plans, or product specifications. It checks adjacency and Boolean profile flags, so it cannot verify room sizes, door swings, visibility angles, anti-ligature hardware, ventilation rates, pressure relationships, travel distance, or authority-specific interpretations. These items require professional review and richer geometric data.

A related limitation is that candidate plans are clean graphs perturbed by the same rule vocabulary used by the checker. This is appropriate for measuring whether the checking layer responds to explicit graph changes, but it is easier than reviewing real designer-produced test-fits. Real plans contain partial adjacencies, diagonal visual relationships, shared vestibules, swing rooms, and renovation constraints that are hard to reduce to a single edge or flag. The next evaluation should connect the checker to a CAD- or BIM-derived graph so that warnings are generated from actual drawings.

The fourth limitation is the external profile library. The profiles are deliberately simplified and use publicly describable logic. They do not represent any jurisdiction's complete regulatory amendments and are not a substitute for licensed professional judgment. A real implementation must be maintained by qualified code consultants and checked with the state health department or authority having jurisdiction.

Finally, the study does not measure downstream design quality, construction cost, schedule impact, or clinician satisfaction after using the tool in a live project. The empirical evidence is limited to translation accuracy and violation-detection behavior. Those measurements are necessary first steps because an unreliable translator or checker would undermine any user-facing system. A later study should embed the workflow in a design charrette, compare review time with and without the checker, and measure whether teams identify more clinically important issues before schematic design.

Conclusion

This paper reframed language-guided healthcare design from automatic hospital floor-plan generation to requirements translation and constraint checking. The distinction is necessary because healthcare design constraints are jurisdictional, clinical, and risk-based. The HPEV-401 experiments show that a structured hybrid translator can recover room-program, adjacency, and risk constraints from source-grounded emergency and behavioral health planning statements with held-out micro-F1 values of 0.874, 0.744, and 0.762. The graph-based checker detects injected test-fit violations with complete recall when supplied with gold constraints and with 0.956 recall at the highest perturbation severity when supplied with Hybrid-predicted constraints. These results support the practical position that LLMs should not be used as autonomous code-compliance engines for hospitals. They should be used as schema-constrained translators whose outputs are routed through auditable external profiles and deterministic checkers. The resulting violation reports give design teams a structured way to discuss clinical intent, profile assumptions, and early test-fit conflicts before schematic design is locked.

The most important lesson is architectural rather than algorithmic: a healthcare design assistant becomes safer when language interpretation, profile authority, and spatial checking are separate modules.

References

[1] International Health Facility Guidelines, Part B: Health Facility Briefing & Design, 75 Emergency Unit and Urgent Care Unit. International Health Facility Guidelines, 2025.

[2] Department of Human Services, Victoria, Design Guidelines for Hospitals and Day Procedure Centres: Standard Components Room Data Sheets. Melbourne, Australia: Department of Human Services, 2004.

[3] U.S. Department of Veterans Affairs, PG-18-9 Space Planning Criteria, Chapter 256: Emergency Department (ED). Washington, DC, USA: Office of Construction and Facilities Management, 2022.

[4] Facility Guidelines Institute, Design of Behavioral Health Crisis Units. St. Louis, MO, USA: Facility Guidelines Institute, 2022.

[5] Facility Guidelines Institute, Guidelines for Design and Construction of Hospitals. St. Louis, MO, USA: Facility Guidelines Institute, 2022.

[6] Facility Guidelines Institute, Guidelines for Design and Construction of Outpatient Facilities. St. Louis, MO, USA: Facility Guidelines Institute, 2022.

[7] ASHRAE, ANSI/ASHRAE/ASHE Standard 170-2021: Ventilation of Health Care Facilities. Atlanta, GA, USA: ASHRAE, 2021.

[8] Centers for Disease Control and Prevention, Guidelines for Environmental Infection Control in Health-Care Facilities. Atlanta, GA, USA: CDC, 2003.

[9] The Joint Commission, National Patient Safety Goal for Suicide Prevention. Oakbrook Terrace, IL, USA: The Joint Commission, 2019.

[10] U.S. Department of Veterans Affairs, Emergency Department Design Guide, PG-18-12. Washington, DC, USA: Office of Construction and Facilities Management, 2021.

[11] New York State Office of Mental Health, Patient Safety Standards, Materials and Systems Guidelines. Albany, NY, USA: NYS Office of Mental Health, 2017.

[12] R. S. Ulrich, C. Zimring, X. Zhu, J. DuBose, H.-B. Seo, Y.-S. Choi, X. Quan, and A. Joseph, "A review of the research literature on evidence-based healthcare design," HERD: Health Environments Research and Design Journal, vol. 1, no. 3, pp. 61-125, 2008.

[13] A. Joseph and M. Rashid, "The architecture of safety: Hospital design," Current Opinion in Critical Care, vol. 13, no. 6, pp. 714-719, 2007.

[14] A. M. Cox, "Behavioral design strategies: Providing a safe and therapeutic ED environment," Health Facilities Management, vol. 31, no. 10, pp. 34-38, 2018.

[15] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.

[16] J. S. Gero, "Design prototypes: A knowledge representation schema for design," AI Magazine, vol. 11, no. 4, pp. 26-36, 1990.

[17] A. Vaswani et al., "Attention is all you need," in Proc. 31st Int. Conf. Neural Information Processing Systems, Long Beach, CA, USA, 2017, pp. 6000-6010.

[18] Chenyu Li, Wenhao Su, and Eric Zhang, “Lightweight Hallucination Firewall for Enterprise LLM Applications: Evidence Consistency, Self-Checking, and Small-Model Detection on TruthfulQA”, JACS, vol. 3, no. 1, pp. 49–65, Jan. 2023, doi: 10.69987/JACS.2023.30104.

[19] Qiyou Wu, Jingwen Bai, and Xiaohan Zhou, “Evidence-Grounded Financial RAG: Reducing Numerical Hallucination in LLM-Generated Corporate Risk Memos”, JACS, vol. 3, no. 3, pp. 65–84, Mar. 2023, doi: 10.69987/JACS.2023.30306.

[20] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proc. 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871-7880.

[21] A. Kalervo, J. Ylioinas, M. Haikio, A. Karhu, and J. Kannala, "CubiCasa5K: A dataset and an improved multi-task model for floorplan image analysis," in Proc. Scandinavian Conf. Image Analysis, 2019, pp. 28-40.

[22] Q. Chen, Q. Wu, R. Tang, Y. Wang, S. Wang, and M. Tan, "Intelligent Home 3D: Automatic 3D-house design from linguistic descriptions only," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops, 2020, pp. 1260-1269.

[23] N. Nauata, K.-H. Chang, C.-Y. Cheng, G. Mori, and Y. Furukawa, "House-GAN: Relational generative adversarial networks for graph-constrained house layout generation," in Computer Vision - ECCV 2020, Cham, Switzerland: Springer, 2020, pp. 162-177.

[24] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[25] Yunhe Li, “Execution-Feedback and Retrieval-Augmented Generation for Conversational Text-to-SQL: From One-Shot Questions to Clarification-Driven Executable Dialogs”, JACS, vol. 3, no. 2, pp. 1–17, Feb. 2023, doi: 10.69987/JACS.2023.30201.