GReFEM: Multimodal LLMs as Zero-Shot Semantic Assistants for Physics-Guided 3D Mesh Refinement

Kartik Bali
Institute of Material Systems Modeling
Helmholtz Zentrum Hereon
Max-Planck-Straße 1, 21502 Geesthacht
kartik.bali@hereon.de

Mahish K.Guru
Institute of Material and Process Design
Helmholtz Zentrum Hereon
Max-Planck-Straße 1, 21502 Geesthacht
mahish.guru@hereon.de

Christian J Cyron
Institute of Material Systems Modeling
Helmholtz Zentrum Hereon
Max-Planck-Straße 1, 21502 Geesthacht
christian.cyron@hereon.de

Roland Aydin
Data-Driven Simulation and Analysis in Materials Science
German Research Center for Artificial Intelligence
Campus D3 2, 66123 Saarbrücken
roland.aydin@dfki.de

Abstract

Adaptive volumetric finite element meshing is a critical step in computer-aided engineering and analysis that dictates the computational budget of a given problem. It traditionally requires iterative PDE solvers or heavily supervised, data-driven surrogates trained on large-scale simulation data. While Multimodal Large Language Models (MLLMs) excel in 2D visual tasks, their zero-shot capability to semantically ground regions based on geometric understanding and physics remains an open question. This work explores a significant question: can the high-level semantic understanding of off-the-shelf MLLMs serve as a viable, zero-shot geometric proxy for finite element mesh refinement? We introduce **GReFEM** (Geometric Reasoning Enhanced Multimodal LLMs for Finite Element Meshing), a framework that utilizes MLLMs to visually localize stress-critical regions based on physics-guided textual prompts. To bridge the gap between 2D MLLM pre-training and 3D geometries, we introduce *orthoViews*, a trained view-selection module that assists these MLLMs by maximizing the observability of key geometric features. We evaluate **GReFEM** by computing finite-element error at stress concentrations, on diverse CAD mesh geometries under multiple loading cases, with MLLMs and prompt variants, against blind and load-informed geometric heuristics, solver-informed Zienkiewicz–Zhu (ZZ) refinement, and a retrained learned adaptive-meshing baseline, all at a matched element budget. Without any FEM solve or simulation training data, **GReFEM** lowers the critical-region peak von Mises stress error relative to a blind geometric heuristic and to refinement driven by a coarse FEM solve, closing 60% of the gap between the unrefined mesh and a reference-solve oracle, while a hand-derived analytical load-informed heuristic remains stronger and the learned baseline performs on par in peak-stress error. With this we investigate their performance as semantic assistants in automated simulation workflows. Code is available at https://github.com/kbali1297/GReFEM.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

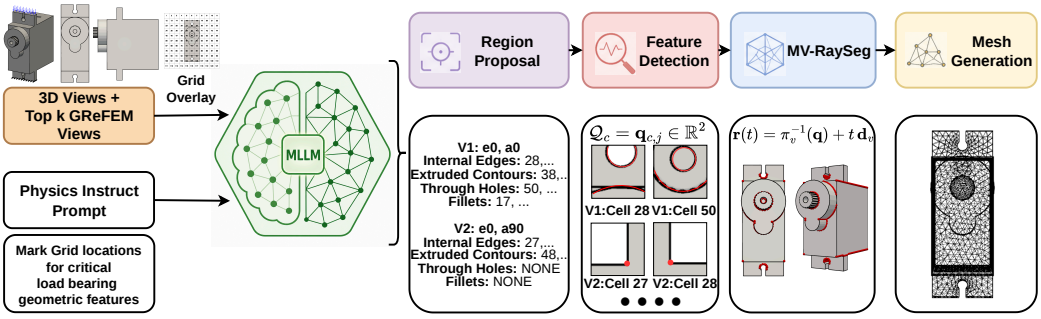


Figure 1: **GReFEM Pipeline**: We process the top views $\mathcal{V}^*$, via the **Region Proposal** and **Feature Detection** Pipelines. Once the points Q are obtained on the views, we use our **Surface Projection (2D to 3D)** pipeline to obtain anchor points $\mathcal{P}$ for mesh generation.

1 Introduction

Recent large language models (LLMs) have demonstrated strong capabilities in reasoning and visual perception, enabling their application across scientific domains that require structured spatial understanding. In particular, recent work has explored the use of LLMs for 3D CAD modeling, object generation, and geometric reasoning, leading to notable advances in design automation and shape synthesis Kienle et al. [2025], Zhou et al. [2025], Yang et al. [2023].

Despite these advances, the role of LLMs in fine-grained numerical simulation workflows remains largely unexplored. High-fidelity finite element simulation fundamentally relies on discretization of the problem domain, where mesh resolution must be adaptively concentrated in regions of high physical variation. While existing approaches adapt LLMs to reason about surface meshes Wang et al. [2024], Fang et al. [2025] and CAD representations Wu et al. [2023], Alrashedy et al. [2024] of 3D objects, comparatively little work investigates their effectiveness in guiding volumetric discretization required to resolve localized physical phenomena while controlling computational cost.

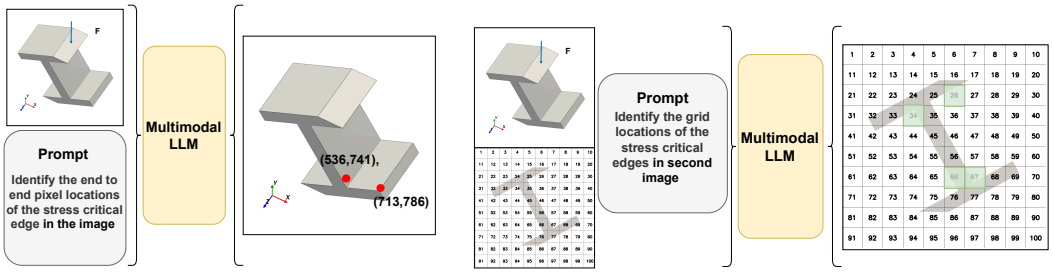
Traditional adaptive volumetric meshing methods rely on PDE solvers to compute error indicators that identify regions requiring refinement (Zienkiewicz and Zhu [1992], Babuška and Rheinboldt [1978]). This tightly couples meshing to solver availability and computational expense. In practice, however, experienced engineers frequently anticipate regions requiring refinement directly from geometric structure, loading directions, and boundary conditions, without explicitly solving for field quantities. Several works exploit such geometry-driven heuristics Bronson et al. [2012], Gkrispanis [2024], but typically assume structured CAD representations that explicitly encode high-level geometric features.

In this work, we investigate a fundamental question: *can off-the-shelf MLLMs act as zero-shot semantic assistants to emulate expert, geometry-driven meshing intuition?* To systematically explore this, we introduce **GReFEM** as a framework to evaluate MLLM spatial grounding in 3D engineering environments. Because MLLMs often struggle with precise pixel-level localization (Fig. 2), we propose grid-based region localization as a solution and integrate *orthoViews*, a trained view-scoring module to maximize geometric observability. This enables us to reliably project coarse 2D MLLM inferences back to the 3D domain, steering volumetric mesh refinement without PDE solvers or explicit CAD annotations.

Through rigorous experiments across CAD geometries, multiple loading conditions, and state-of-the-art MLLMs, we present the following contributions:

(1) **A Semantic Evaluation Framework:** We introduce **GReFEM** (with *orthoViews*) as a CAD-agnostic probing pipeline that translates geometric instructions grounded in problem physics into localized 3D volumetric mesh refinement anchors. We score every refinement policy by the downstream finite-element error it produces at a matched element budget, namely the peak von Mises stress and the strain energy inside the stress-critical region of a high-fidelity reference solve.

(2) **Analysis of Instruction-Driven Grounding:** Prompt content measurably steers where MLLMs refine. Describing only the load (*Load-Only*) raises recall and anchor count but lowers anchor



(a) Pixel localization.

(b) Region (grid-cell) localization.

Figure 2: **Localization behavior of MLLMs** *Left:* Direct pixel-level localization is unreliable, even when the model correctly identifies the stress-critical feature semantically. *Right:* Grid-based region localization in **GReFEM** views yields slightly more accurate results: Predicted green grid cells are concentrated around stress-critical re-entrant edges.

precision for all four MLLMs, and at a matched element budget enumerating the stress-critical feature types lowers peak-stress error relative to *Load-Only* for Gemini-3-flash and GPT-5.4-mini.

(3) **Downstream Accuracy at a Matched Budget:** Across multiple geometry-load cases, zero-shot **GReFEM** yields lower critical-region peak-stress error than a blind geometric heuristic and ZZ refinement from a coarse FEM solve, with no solve of its own, and performs comparably overall to a learned-AMR baseline AMBER (Freymuth et al. [2025]) retrained on 440 expert meshes without requiring any of them. A load-informed heuristic built from hand-derived beam-theory formulas remains stronger, but relies on hand-derived analytical form for every separate loading case contrary to **GReFEM**, which treats loading cases as different only at the prompt level.

2 Related Works

Adaptive Meshing Adaptive meshing non-uniformly allocates resolution to minimize finite element simulation costs. Traditional pipelines rely on computationally expensive, solver-in-the-loop error estimators tailored to the governing PDEs (Babuška and Rheinboldt [1978], Babuvška and Rheinboldt [1978], Zienkiewicz and Zhu [1992]), though some geometry-driven heuristics avoid explicit PDE solves using mathematical criteria (Rivara [1984], Shewchuk [2002], Bronson et al. [2012]). Recently, deep learning has emerged as a surrogate, using Graph Neural Networks (GNNs) to learn h-adaptive sizing fields (Khan et al. [2024], Freymuth et al. [2025]) or r-adaptive topological optimizations (Pfaff et al. [2020], Rowbottom et al. [2024]). While these supervised architectures achieve high accuracy, their strict reliance on massive datasets of converged FEM solutions prevents zero-shot generalization. Rather than proposing a data-hungry supervised model, **GReFEM** explores a new solver-free paradigm: evaluating whether the semantic reasoning of off-the-shelf MLLMs can act as a natural language-guided proxy for mesh refinement, significantly circumventing the need for offline FEM training data. Furthermore, supervised GNNs fundamentally solve a different problem: they act as end-to-end surrogates for a specific, pre-defined PDE and boundary condition distribution. Adapting a GNN to a new physical phenomenon (e.g., thermal stress instead of linear elasticity) or a fundamentally different load type requires complete model retraining and a newly generated FEM dataset. In contrast, **GReFEM** acts as a domain-agnostic semantic interface. By shifting the refinement logic from learned network weights to expert-provided natural language prompts, **GReFEM** can dynamically adapt to entirely new loading scenarios without a single parameter update.

3D Spatial Reasoning with MLLMs Extending vision-language models to 3D reasoning remains challenging due to limited 3D training data. Several approaches address this by directly training LLMs on structured 3D representations, such as 3D feature fields Hong et al. [2023], point clouds (Xu et al. [2024], Batra et al. [2025]), or radiance fields Kerr et al. [2023]. Alternatively, other works infer 3D understanding from 2D projections without explicit 3D structures. This includes predicting 3D bounding boxes Cho et al. [2024], optimizing view selection for 3D QA Wang et al. [2025], regressing geometric attributes via 3D query tokens Yang et al. [2026], estimating absolute distances from synthetic datasets Chen et al. [2024], progressive prompting for trajectory imagination Ma et al. [2024], and leveraging video sequences for 3D grounding Zheng et al. [2025]. Our work follows

this latter 2D-to-3D projection paradigm; however, rather than targeting general semantic scene understanding, we uniquely leverage intelligently ranked 2D orthographic views to infer mechanically relevant 3D features for downstream engineering discretization.

3 Preliminaries

Governing Equations We consider linear elastic deformation of a solid body occupying a bounded domain $\Omega \subset \mathbb{R}^3$, subject to prescribed boundary displacements. Let $\mathbf{u} : \Omega \to \mathbb{R}^3$ denote the displacement field, which satisfies

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) &= \mathbf{0} \quad \text{in } \Omega, \\ \mathbf{u} &= \bar{\mathbf{u}} \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\bar{\mathbf{u}}$ denotes the prescribed boundary displacements. In all experiments, loads are induced by imposing fixed displacements on opposing boundaries of the domain.

Downstream Error Metrics Closed-form solutions are unavailable for these geometries, so we assess a candidate mesh by how closely its finite-element solution $\mathbf{u}_h$ reproduces the high-resolution reference $\mathbf{u}_{\mathrm{ref}}$ on the design-governing quantities, evaluated *inside* the stress-critical region that the refinement budget targets rather than over the smooth bulk.

Critical region. Let $\eta_{ZZ}(\mathbf{x})$ be the Zienkiewicz–Zhu error indicator (Appendix A.1) of the reference solution and q_p its p-th percentile over Ω. The reference's most stress-critical points are the top 0.1% of this field, $\mathcal{C} = \{\mathbf{x} \in \Omega : \eta_{ZZ}(\mathbf{x}) \geq q_{99.9}\}$ (after excluding a 2% slab of the bounding box at the loaded boundaries, which removes load-application singularities), and the critical region Ω_{crit} is their volumetric neighbourhood of radius $r_{\mathrm{crit}} = 4\, h_{\min}$:

$$\Omega_{\mathrm{crit}} = \Big\{ \mathbf{x} \in \Omega \;:\; \min_{\mathbf{c} \in \mathcal{C}} \|\mathbf{x} - \mathbf{c}\|_2 \leq r_{\mathrm{crit}} \Big\}. \tag{2}$$

Peak von Mises stress. The failure-relevant quantity is the von Mises stress, built from the deviatoric part $\mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\sigma})\,\mathbf{I}$ of the Cauchy stress $\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}(\mathbf{u})$,

$$\sigma_{vM}(\mathbf{u}) = \sqrt{\tfrac{3}{2}\,\mathbf{s}(\mathbf{u}) : \mathbf{s}(\mathbf{u})}. \tag{3}$$

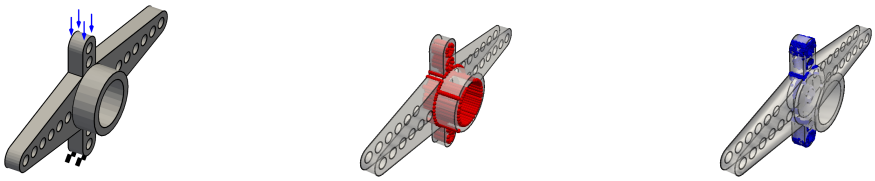
To remain robust to the integrable singularities at re-entrant corners we summarise it by its 99-th percentile over the critical region, $\sigma_{vM}^{p99}(\mathbf{u}) = \mathrm{P}_{99}\{\sigma_{vM}(\mathbf{u}(\mathbf{x})) : \mathbf{x} \in \Omega_{\mathrm{crit}}\}$, and report the relative error against the reference,

$$\delta_{vM}^{p99} = \frac{\big|\, \sigma_{vM}^{p99}(\mathbf{u}_h) - \sigma_{vM}^{p99}(\mathbf{u}_{\mathrm{ref}}) \,\big|}{\sigma_{vM}^{p99}(\mathbf{u}_{\mathrm{ref}})}. \tag{4}$$

Critical-region strain energy. As the natural norm of the elasticity operator we also report the strain energy stored in Ω_{crit} and its relative error,

$$E_{\mathrm{crit}}(\mathbf{u}) = \tfrac{1}{2}\int_{\Omega_{\mathrm{crit}}} \boldsymbol{\varepsilon}(\mathbf{u}) : \mathbb{C} : \boldsymbol{\varepsilon}(\mathbf{u})\, \mathrm{d}\Omega, \qquad \delta_E^{\mathrm{crit}} = \frac{\big|\, E_{\mathrm{crit}}(\mathbf{u}_h) - E_{\mathrm{crit}}(\mathbf{u}_{\mathrm{ref}}) \,\big|}{E_{\mathrm{crit}}(\mathbf{u}_{\mathrm{ref}})}. \tag{5}$$

$(\delta_{vM}^{p99},\ \delta_E^{\mathrm{crit}})$ (both relative, lower is better) are the two quantities reported throughout (Table 1); restricting them to Ω_{crit} prevents the large, well-resolved smooth region from masking the differences at the concentrations that refinement must resolve. Ω_{crit} is a fixed region of space, identical for every candidate mesh, and each quantity is evaluated natively on the candidate's own mesh, so no cross-mesh interpolation enters. Because σ_{vM}^{p99} sits at stress concentrations that do not converge under h-refinement, absolute errors stay high for every policy; the numbers are meaningful for *ranking* policies at a common budget, not as convergence estimates. We aggregate over cases by the mean and median and compare policies within each geometry and load case.



(a) Compression Loading Case (b) **GReFEM** Predicted refinement anchors (c) Ground-truth ZZ stress indicator.

Figure 3: Qualitative evaluation of **GReFEM**. Predicted refinement points (Red) align with stress-critical regions (Blue) identified by the ground truth ZZ stress error indicator field computed using the fine reference solution u_{ref}. Blue points within the geometry denote stress critical areas above the 99th percentile of the η_{ZZ} field.

Finite Element Meshing We employ tetrahedral volumetric meshes due to their ability to conform to complex 3D geometries while maintaining element quality under local refinement. Mesh generation is performed using Gmsh (Geuzaine and Remacle [2009]), where spatially varying element sizes are prescribed via a distance-based sizing field. Given a set of refinement anchor points $\mathcal{P} \subset \mathbb{R}^3$ located on the surface, we define a distance field $d(\mathbf{x}) = \min_{\mathbf{p}\in\mathcal{P}} \|\mathbf{x} - \mathbf{p}\|_2$, which measures the proximity of any point $\mathbf{x} \in \Omega$ to the closest refinement location. This distance field is mapped to a spatially varying linear target element size function $h(\mathbf{x})$,

$$h(\mathbf{x}) = \begin{cases} h_{\min}, & d(\mathbf{x}) \le D_{\min}, \\ h_{\max}, & d(\mathbf{x}) \ge D_{\max}, \\ \ell(d(\mathbf{x})), & \text{otherwise}, \end{cases} \tag{6}$$

$$\ell(d) = h_{\min} + \frac{d - D_{\min}}{D_{\max} - D_{\min}} (h_{\max} - h_{\min}). \tag{7}$$

4 Method

We propose **GReFEM**, a semantic spatial grounding testbed framework applied to adaptive finite element meshing. Rather than relying on solver-based error estimators, **GReFEM** leverages pretrained MLLMs and problem specific instructions to identify mechanically salient regions using only visual observations of geometry. **GReFEM** operates solely on triangulated surface meshes and requires no access to B-Rep entities, STEP files, or simulation feedback.

4.1 Problem Formulation

We consider a solid domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$ represented by a triangulated surface mesh $\mathcal{M} = (V, F)$. The objective is to generate a volumetric finite element mesh that locally refines regions expected to exhibit large stress gradients under mechanical load, while maintaining coarse resolution elsewhere to control computational cost. Given $\mathcal{M}$, the method predicts a sparse set of surface-localized refinement points $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$, corresponding to geometrically or mechanically important features, such as sharp edges, fillets, holes and load-bearing faces. The resulting size field $h : \Omega \to \mathbb{R}^+$, as constructed according to equation 6, is provided to a standard tetrahedral meshing algorithm in Gmsh, which deterministically generates a conforming volumetric mesh.

4.2 orthoViews

We render a candidate set of V orthographic views $\mathcal{V} = \{v_k = (\theta_k, \phi_k)\}_{k=1}^V$ by systematically enumerating azimuth and elevation angles on a fixed angular grid. Orthographic view rendering

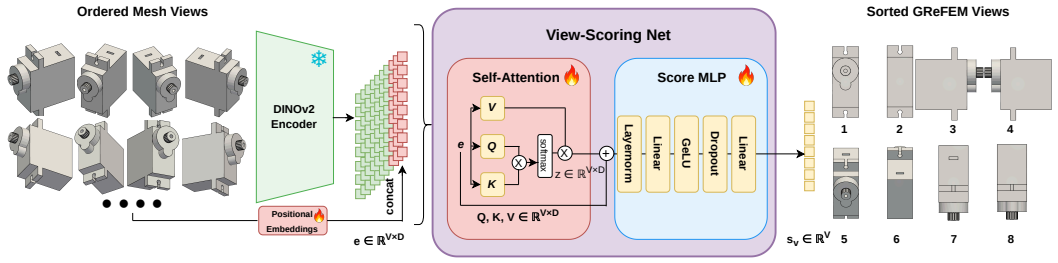


Figure 4: *orthoViews* performs self-attention across the V views of each CAD geometry, allowing each view embedding to aggregate information from all other views before being scored by a per-view MLP.

preserves metric structure and avoids perspective distortion, which is critical for our projection pipeline. Each view is embedded using a frozen DINOv2 encoder, yielding feature vectors $\mathbf{e}_k \in \mathbb{R}^D$. To score views by geometric informativeness, we order these, compute self-attention across the views V and input these view embeddings to the *Score-MLP* network, in Figure 4. This ranking module is trained using a pairwise margin loss to prioritize views that expose a higher number of salient geometric features, given that we have view labels. The top-K views form the final inference set $\mathcal{V}^*$ for **GReFEM**.

4.3 Inference Pipeline

Figure 1 summarizes the **GReFEM** inference pipeline, which consists of three stages: region proposal, feature detection, and surface projection.

Region Proposal Each selected view $v \in \mathcal{V}^*$ is overlaid with a uniform grid $\mathcal{G}_v$. A multimodal LLM predicts a subset of salient grid cells $\mathcal{C}_v \subset \mathcal{G}_v$, corresponding to geometrically or mechanically significant regions. Proposals are aggregated across views via $\mathcal{C} = \bigcup_{v \in \mathcal{V}^*} \mathcal{C}_v$

Feature Detection Each cell in $\mathcal{C}$ defines an image patch processed using classical vision operators and mesh-ray projection to recover precise feature locations in the 2D views. We specifically spot features resembling internal corner edges, extruded contours/bosses, holes and fillets that are key geometric discontinuities in a load bearing structure. This stage converts coarse LLM output into accurate 2D feature pixels $\mathcal{Q}_c = \{\mathbf{q}_{c,j} \in \mathbb{R}^2\}$.

MV-RaySeg (Multi-View Ray Segment Sampling) Given a feature point $\mathbf{q} \in \mathbb{R}^2$ detected in a view v, we recover its corresponding spatial projection on the 3D surface by ray–mesh intersection under known camera parameters. Let π_v^{-1} denote the inverse projection for view v, and let $\mathbf{d}_v \in \mathbb{R}^3$ be the associated view direction. The corresponding ray is defined as $\mathbf{r}(t) = \pi_v^{-1}(\mathbf{q}) + t\,\mathbf{d}_v, \quad t \in \mathbb{R}$.

We compute the complete ordered set of ray–mesh intersections with the surface mesh $\mathcal{M}$, $\mathcal{T}_q = \{t_i \mid \mathbf{r}(t_i) \in \mathcal{M}\}$, with $t_1 < t_2 < \cdots < t_{|\mathcal{T}_q|}$. Under standard assumptions on watertight meshes, consecutive intersection pairs (t_{2k-1}, t_{2k}) define ray segments that lie inside the solid. For each such segment, we sample a finite set of 3D points $\mathcal{P}_q = \bigcup_k \{\mathbf{r}(t) \mid t \in [t_{2k-1}, t_{2k}]\}$, which yields a spatially extended set of surface-adjacent anchor points corresponding to the 2D feature $\mathbf{q}$. This formulation allows a single 2D detection to influence multiple depth-aligned surface regions, which is essential for internal cavities and edges.

5 Experiments

Training We train the *orthoViews* view-selection module on a dataset consisting of 5k diverse CAD parts from ABC CAD dataset Koch et al. [2019] and FreeCAD Riegel et al. [2016] spanning mechanical components with varying geometric complexity (e.g., extrusions, holes, fillets, and load-bearing features). Each object is represented during training by both its triangulated surface mesh and the corresponding STEP file. Specifically, we leverage high-level geometric primitives available in STEP (e.g., planar faces, cylindrical axes) for training by scoring each view by the number of features aligned in its view direction. During inference, our framework can work *solely* with spherical views

of surface meshes of 3D objects. For more information, the reader is encouraged to read the appendix section on *orthoViews*. Training the network on the entire dataset for 4 epochs took a couple of hours on one NVIDIA H100 GPU.

Experimental Setup We evaluate on 156 CAD parts from the ABC Koch et al. [2019] and FreeCAD Riegel et al. [2016] datasets (Appendix B), disjoint from the *orthoViews* training and validation splits, each under five loading scenarios, uniaxial (compression, bending, torsion) and multi-axial (bending-compression, torsion-compression), giving 780 paired geometry–load cases. We use three prompt templates: *Load+Features*, *Load+Features+Heuristics* and *Load-Only*. For the *Load+Features* prompt we provide the model with the loading instruction prompt and the key geometric features that could be stress critical under the load (Internal corner Edges, Corner Edges of extruded or Protruding parts, Holes and Fillets). The model is tested on the ability to correctly identify the cells under each feature enough to localize it using contour and depth detection CV pipeline. For the *Load+Features+Heuristics* we provide additional load based hints to spot stress critical regions in addition to the input provided in the *Load+Features* prompt. For the *Load-Only* prompt, we merely describe the loading on the object and instruct the MLLM to recommend cells containing the stress-critical geometric features. All the cells obtained are given to contour and depth detection algorithms in feature detection part of the pipeline to compute anchor points in these 2D views that will be later projected onto the 3D mesh. We evaluate four MLLMs, Gemini-3-flash-preview, Claude-Haiku-4.5, GPT-5.4-mini and Qwen3-VL-235B, all accessed through the OpenRouter (OpenRouter, Inc. [2026]) API and all given the top five *orthoViews* views, without domain-specific training or PDE solves. The minimum element size is $h_{\min} = 0.2\, h_{\max}$ for all refined meshes, and $D_{min} = 2h_{min}$, $D_{max} = 4h_{min}$ define the element size field of Eq. 6. Anchors within $6h_{\min}$ of the loaded faces are ignored by the mesher for every policy alike, so that error differences arise from interior refinement and not from differences in the imposition of boundary conditions. Every (geometry, load) case is scored against an independent reference solve on a uniform mesh with $h_{\text{fine}} = h_{\max}/16$, capped at 6×10^6 tetrahedra ($\approx 5 \times 10^6$ typical; linear elasticity, $E = 210\,\text{GPa}$, $\nu = 0.3$), through the critical-region errors of Eqs. 4 and 5.

Qualitative Results In the qualitative examples, the **GReFEM** predicted refinement locations align with stress-critical regions. Notably, the predicted points are not uniformly distributed across all geometric features such as holes or internal corners, but are selectively localized to those features that should be mechanically activated under the prescribed loading conditions (for instance Figure 3c). In particular, while multiple concave features may be present in the geometry, **GReFEM**, when provided with physics guided prompts about the problem and physics, attempts to strongly highlight those regions that correspond to dominant stress concentrations in the ZZ stress field, rather than marking all such features indiscriminately, as seen in Figure 6 in the appendix.

Refinement Policies at a Matched Budget Adaptive meshing is fundamentally a resource-constrained optimization problem: introducing too many refinement points severely degrades downstream solver efficiency. We therefore compare seven refinement policies at a *matched element budget*, so that differences reflect *where* smaller elements are placed rather than how many are spent: (i) **Coarse**, the unrefined mesh produced by the pipeline (a fixed floor, not budget-matched); (ii) **GReFEM**, our framework with trained *orthoViews* selection and zero-shot MLLM inference; (iii) **Geometric heuristic (blind)**, the load-agnostic detector of concave sharp edges, fillets, and holes based on dihedral angles and discrete curvature, independent of loading; (iv) **Load-informed heuristic**, the same geometric candidates re-weighted by a solver-free closed-form nominal-stress estimate (per-layer voxel cross-section properties A, I_z, J along the loading axis), so the loading case decides which candidates are refined; (v) **ZZ from coarse solve**, anchors placed at the top-percentile ZZ error points of a single coarse FEM solve; (vi) **ZZ from fine reference solve**, anchors at the top-percentile ZZ points of the $\approx 5 \times 10^6$-element reference solution, an oracle reference; and (vii) **AMBER** Freymuth et al. [2025], a learned sizing-field AMR method retrained from scratch with load-case conditioning on 440 expert meshes derived from reference solves. Each anchor-based baseline is reduced to **GReFEM**'s anchor count for that object, and the $h_{\min}$ of every refined baseline is calibrated iteratively to **GReFEM**'s element count (10% tolerance); AMBER emits a continuous field, matched via a global scaling factor. The load-informed heuristic and AMBER are detailed in Appendix A.3.

Table 1: **Downstream FEM error at a matched element budget** on 156 CAD parts $\times$ 5 load cases. Relative errors in peak von Mises stress (δ_{vM}^{p99}, Eq. 4) and in strain energy (δ_E^{crit}, Eq. 5) inside Ω_{crit} against the $\approx 5 \times 10^6$-element reference solve (lower is better). The fine-reference ZZ policy is an oracle reference; all other refined policies are matched to GReFEM's anchor count and element budget. †Budget-matched to GReFEM (the native *Load-Only* runs of Gemini and GPT place more anchors). ‡Not budget-matched (native element count). AMBER fails to mesh $\approx 6\%$ of cases (735 cases evaluated).

Method	**Information used**	δ_{vM}^{p99} **mean**	δ_{vM}^{p99} **med.**	δ_E^{crit} **mean**	δ_E^{crit} **med.**	**cells**
Coarse (no refinement)	—	0.739	0.786	0.944	0.967	24 660
GReFEM (Gemini)	Load + Features + Heuristics	0.626	0.649	0.879	0.919	95 052
GReFEM (GPT-5.4-mini)	Load + Features + Heuristics	0.630	0.651	0.873	0.915	96 481
GReFEM (Claude-Haiku-4.5)	Load + Features + Heuristics	0.643	0.652	0.880	0.925	87 216
GReFEM (Qwen3-VL-235B)	Load + Features + Heuristics	0.651	0.672	0.886	0.924	82 774
GReFEM (Gemini)	Load + Features	0.634	0.665	0.872	0.920	97 095
GReFEM (GPT-5.4-mini)	Load + Features	0.641	0.663	0.878	0.920	92 620
GReFEM (Claude-Haiku-4.5)	Load + Features	0.634	0.654	0.877	0.920	95 691
GReFEM (Qwen3-VL-235B)	Load + Features	0.646	0.675	0.889	0.930	84 970
GReFEM (Gemini)†	Load-Only	0.649	0.675	0.881	0.923	93 715
GReFEM (GPT-5.4-mini)†	Load-Only	0.666	0.693	0.890	0.925	93 226
GReFEM (Claude-Haiku-4.5)‡	Load-Only	0.637	0.660	0.871	0.914	112 773
GReFEM (Qwen3-VL-235B)‡	Load-Only	0.639	0.656	0.878	0.915	91 521
Geometric heuristic (blind)	geometry only	0.672	0.696	0.892	0.928	93 683
Load-informed heuristic	geometry + hand-derived closed-form stress	0.567	0.591	0.869	0.914	87 746
ZZ from coarse solve	1 FEM coarse solve + ZZ estimator	0.672	0.700	0.887	0.925	92 249
AMBER (learned AMR)	supervised expert meshes + load	0.625	0.660	0.869	0.915	84 856
ZZ from fine reference solve	1 FEM reference solve + ZZ estimator	*0.549*	*0.578*	*0.850*	*0.908*	*81 929*

5.1 Downstream FEM Error at a Matched Budget

We treat downstream FEM error as the primary criterion. Table 1 reports the relative error in peak von Mises stress (δ_{vM}^{p99}, the failure-relevant design quantity) and in δ_E^{crit} (the natural finite-element error norm), both inside Ω_{crit}, pooled over all 780 cases.

GReFEM versus solver-free, solver-informed and learned baselines. With no FEM solve, no simulation training data and no hand-derived mechanics, **GReFEM** (Gemini, *Load+Features+Heuristics*) attains a lower peak-stress error than the blind geometric heuristic (pooled 0.626 vs. 0.672) and than ZZ refinement from a coarse FEM solve (0.672), closing 60% of the gap between the coarse mesh and the reference-solve oracle at a forward cost of view rendering plus one MLLM call (median ≈ 7 s). Strain-energy errors separate the refined policies as well. **GReFEM**'s energy error is on average lower than blind heuristic and coarse-solve ZZ. AMBER, retrained with load conditioning on 440 expert meshes, and **GReFEM** are statistically indistinguishable in peak-stress error (0.625 vs. 0.626), while AMBER attains a slightly lower strain-energy error (0.869 vs. 0.879). **GReFEM** reaches this range with no expert meshes, no reference solves and no meshing failures. While the load-informed heuristic surpasses **GReFEM** in peak-stress error (0.567 vs. 0.626), it requires an expert to hand-derive beam-theory stress formulas for each load family and is valid only where nominal-stress reasoning applies, whereas **GReFEM** adapts to a new load family by a prompt edit. We report it as the boundary of what semantic localization currently provides.

MLLMs and prompts. With the feature-enumerating prompts all four MLLMs lie within 0.025 in pooled δ_{vM}^{p99}. At a matched budget, these prompts lower peak-stress error relative to *Load-Only* for Gemini (0.626 vs. 0.649) and GPT-5.4-mini (0.630 vs. 0.666), whereas adding load-specific heuristics to the feature list (*Load+Features+Heuristics* vs. *Load+Features*) decreases MLLM's F1 while slightly increasing precision. For Claude and Qwen the *Load-Only* runs place more anchors and use 10–30% more elements than the feature prompts, thus displaying lower localization precision. The sizing sweep (Table 2) shows that **GReFEM**'s advantage over the blind heuristic and coarse-solve ZZ is largest at the full-budget operating point and decays toward parity as elements become scarce, while the load-informed heuristic stays ahead at every level.

5.2 Localization Overlap versus Downstream Accuracy

For measuring the degree to which MLLMs localize refinement points based on prompt variations and number of input views, we take the ZZ stress indicator field η_{ZZ} of the reference solve and extract the

points above its 99th percentile (after the same 2% boundary trim as Ω_{crit}). **Recall@r** is the fraction of these reference points within radius r of any predicted refinement anchor, and **Precision@r** the fraction of predicted anchors within r of a reference point; **F1@r** is their harmonic mean. We set $r = D_{max} = 4h_{min}$ (Eq. 6), the extent of the refinement transition zone. Metrics are computed per case with the top five *orthoViews* views and averaged over cases; cases in which a variant returned no anchors ($\leq 9\%$ per variant) are excluded from that variant's average.

Appendix Table 4 reports anchor-level Precision@r, Recall@r and F1@r for all twelve MLLM–prompt variants over the 780 cases. Contrary to common perception, we find the localization overlap metrics are decorrelated with actual downstream FEM accuracy. Reduced to **GReFEM**'s anchor count, the blind geometric heuristic attains higher recall (0.770) and F1 (0.596) than every MLLM variant (F1 0.485–0.579), and its precision (0.571) is matched only by Qwen3-VL-235B (0.572), yet it has a higher peak-stress error than **GReFEM** at the same element budget (Table 1). Conversely, the load-informed heuristic has the lowest recall (0.488) and F1 (0.469) of all policies and the lowest downstream error. Overlap metrics weight every top-percentile ZZ point equally, whereas σ_{vM}^{p99} is governed by the few peak concentrations: a policy can cover many mildly critical points while under-resolving the peak. The quantity that does track the downstream ranking is the element size the final mesh places at the hotspots (Appendix Table 7): median 0.42 (load-informed), 0.45 (GReFEM) and 0.49 (blind heuristic), in the same order as their peak-stress errors. We therefore use localization overlap only to characterise *how* the prompt steers each model in the following section.

5.3 Analysis of Zero-Shot Semantic Grounding

The Impact of Physics-Guided Prompting A central hypothesis of this study is that MLLMs can translate expert-provided textual cues into load-dependent spatial selections. We compare three prompt variants: *Load+Features* (loading conditions plus the stress-critical feature types), *Load+Features+Heuristics* (adding load-specific spatial hints), and *Load-Only* (loading conditions only, asking for "stress critical" regions). The effect of the feature list is consistent across all four MLLMs: without it (*Load-Only*) the models propose more anchors ($+0.7$k to $+2.1$k per case), recall rises (e.g. GPT-5.4-mini $0.644 \rightarrow 0.778$) and precision falls by 0.015–0.037 (Appendix Table 4). Where *Load-Only* can be budget-matched (Gemini, GPT-5.4-mini) the broader selection costs downstream accuracy: peak-stress error rises from 0.626 to 0.649 and from 0.630 to 0.666. Adding load-specific heuristics to the feature list makes the selection slightly more conservative: anchors per case fall by up to 0.6k, precision rises marginally ($+0.001$ to $+0.008$), and recall drops for Gemini, Claude, and Qwen (0.008–0.061; unchanged for GPT-5.4-mini). Downstream, the effect has no consistent sign: peak-stress error falls for Gemini and GPT-5.4-mini (-0.008, -0.011) and rises for Claude and Qwen ($+0.009$, $+0.005$).

Load Cases Per load case (Appendix Table 5, pooled over all 780 cases), **GReFEM** has lower peak-stress error than the blind heuristic in all five load cases and coarse-solve ZZ in all five. The load-informed heuristic's lead is largest under torsion (0.514 vs. 0.622), where the closed-form $\tau = Tr/J$ is most informative. FEM errors vary more across load cases (range 0.063) than across MLLMs (0.018) or prompts (0.005) (Appendix Table 6).

Table 2: **Sizing sweep.** δ_{vM}^{p99} / δ_E^{crit} in Ω_{crit} as a common scale factor is applied to $h_{\min}$ and $h_{\max}$; every baseline is re-matched to **GReFEM**'s element budget at each level (lower is better; 773–780 cases per level). The **GReFEM** column pools all four MLLMs (Gemini, GPT-5.4-mini, Claude-Haiku-4.5, Qwen3-VL-235B) per case (*Load+Features+Heuristics* prompt); the MLLMs agree within ≈ 0.02.

sizing	mean cells	GReFEM	Geom. heur.	Load-inf.	ZZ coarse	AMBER	ZZ fine ref.
$h \times 1.0$	90k	0.637/0.879	0.672/0.892	0.567/0.869	0.672/0.887	0.625/0.869	*0.549/0.850*
$h \times 1.25$	54k	0.689/0.903	0.687/0.895	0.627/0.890	0.717/0.905	0.657/0.891	*0.608/0.869*
$h \times 1.6$	32k	0.728/0.928	0.745/0.934	0.686/0.915	0.728/0.926	0.716/0.927	*0.700/0.915*
$h \times 2.0$	20k	0.749/0.938	0.766/0.943	0.734/0.928	0.760/0.944	0.736/0.931	*0.728/0.931*

Table 3: **Per-case cost**, pooled over all 780 cases. Maximum-displacement error is globally computed and reported for completeness; the decisive column is the anchor-placement cost, where **GReFEM**'s full forward cost is view rendering plus a single MLLM call with *no* FEM solve. The **GReFEM** row averages all four MLLMs (Gemini, GPT-5.4-mini, Claude-Haiku-4.5, Qwen3-VL-235B); the cost profile is MLLM-independent.

method	mean cells	max-disp. err (med.)	solve time (med.)	anchor-placement cost (med.)
Coarse	24 660	0.025	0.2 s	—
GReFEM	90 379	0.023	0.8 s	$\approx$ 7 s: render + 1 MLLM call; **no solve**
Geometric heuristic	93 683	0.022	0.8 s	$<$ 1 s; no solve
Load-informed heuristic	87 746	0.022	1.0 s	$<$ 1 s; no solve
ZZ from coarse solve	92 249	0.021	2.5 s	1 coarse solve + ZZ
ZZ from fine reference solve	81 929	0.021	0.6 s	reference solve ($\approx$ 35–60 min)

6 Conclusion

This study explored a fundamental question at the intersection of foundation models and computational engineering: can MLLMs act as zero-shot semantic assistants that replace solver-in-the-loop decisions about *where* to refine a mesh? Measured by the finite-element error at stress concentrations, the quantity refinement is meant to improve, the answer on 156 parts and 780 geometry–load cases is a qualified yes. At a matched element budget and without any FEM solve or simulation data, **GReFEM** lowers critical-region peak-stress error relative to a blind geometric heuristic and to ZZ refinement from a coarse solve, performs comparably overall to a learned baseline AMBER, and is insensitive to the choice among four MLLMs. The gain comes for certain MLLMs (Gemini, GPT) from following explicit instructions about which feature types are stress critical under the given load: removing that list broadens the selection and, at a matched budget, increases error.

Limitations Our study has several limitations. A load-informed heuristic built from hand-derived beam-theory formulas for each loading case outperforms **GReFEM**, **GReFEM** matches but does not outperform the retrained learned baseline AMBER (peak-stress error 0.626 vs. 0.625), and 40% of the gap to the reference-solve oracle remains. The MLLMs rely on the classical feature-detection stage to turn coarse grid-cell proposals into anchors, and on the presence of recognisable geometric features. **GReFEM** decides only *where* to refine; the refinement magnitude ($h_{\min}$, $D_{\min}$, $D_{\max}$) and the final mesh generation follow fixed rules. Our evaluation covers linear elasticity under five displacement-driven load cases on ABC and FreeCAD parts. Within these bounds, the results indicate that off-the-shelf MLLMs, without task-specific training, can serve within **GReFEM** as solver-free, prompt-conditioned refinement policies and as an interface between human physical intuition and low-level numerical discretization.

Acknowledgements and Disclosures

The authors gratefully acknowledge Helmholtz-Zentrum Hereon for providing the GPU computing resources on which all experiments in this work were carried out, and thank Prof. Christian Cyron, Head of the Institute of Material Systems Modeling (MSM), Helmholtz-Zentrum Hereon, for providing the platform and support that made this work possible. The authors also thank their colleagues at the Institute of Material Systems Modeling for helpful discussions. The authors have no competing interests to declare that are relevant to the content of this article.

References

Claudius Kienle, Benjamin Alt, Darko Katic, Rainer Jäkel, and Jan Peters. Querycad: Grounded question answering for cad models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5798–5805. IEEE, 2025.

Jiwei Zhou, Jorge D Camba, and Pedro Company. Cadialogue: A multimodal llm-powered conversational assistant for intuitive parametric cad modeling. *Computer-Aided Design*, page 104006, 2025.

Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023.

Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. Llama-mesh: Unifying 3d mesh generation with language models. *arXiv preprint arXiv:2411.09595*, 2024.

Shuangkang Fang, I Shen, Yufeng Wang, Yi-Hsuan Tsai, Yi Yang, Shuchang Zhou, Wenrui Ding, Takeo Igarashi, Ming-Hsuan Yang, et al. Meshllm: Empowering large language models to progressively understand and generate 3d mesh. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14061–14072, 2025.

Sifan Wu, Amir Khasahmadi, Mor Katz, Pradeep Kumar Jayaraman, Yewen Pu, Karl Willis, and Bang Liu. Cad-llm: Large language model for cad generation. In *Proceedings of the neural information processing systems conference. neurIPS*, volume 1, 2023.

Kamel Alrashedy, Pradyumna Tambwekar, Zulfiqar Zaidi, Megan Langwasser, Wei Xu, and Matthew Gombolay. Generating cad code with vision-language models for 3d designs. *arXiv preprint arXiv:2410.05340*, 2024.

OC Zienkiewicz and JZ Zhu. The superconvergent patch recovery. *Computer methods in applied mechanics and engineering*, 101(1-3):207–224, 1992.

Ivo Babuška and Werner C Rheinboldt. A-posteriori error estimates for the finite element method. *International journal for numerical methods in engineering*, 12(10):1597–1615, 1978.

Jonathan R Bronson, Joshua A Levine, and Ross T Whitaker. Particle systems for adaptive, isotropic meshing of cad models. *Engineering with computers*, 28(4):331–344, 2012.

Konstantinos Gkrispanis. Enhancing machining feature recognition in cad models through transfer learning on brepnet. In *35. Forum Bauinformatik, fbi 2024*, pages 42–49. Technische Universität Hamburg, Institut für Digitales und Autonomes Bauen, 2024.

Niklas Freymuth, Tobias Würth, Nicolas Schreiber, Balazs Gyenes, Andreas Boltres, Johannes Mitsch, Aleksandar Taranovic, Tai Hoang, Philipp Dahlinger, Philipp Becker, et al. Amber: Adaptive mesh generation by iterative mesh resolution prediction. *arXiv preprint arXiv:2505.23663*, 2025.

I Babuvška and Werner C Rheinboldt. Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4):736–754, 1978.

M Cecilia Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International journal for numerical methods in Engineering*, 20(4):745–756, 1984.

J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22(1-3):21–74, 2002. doi: 10.1016/S0925-7721(01)00071-7.

Ainulla Khan, Moyuru Yamada, Abhishek Chikane, and Manohar Kaul. Graphmesh: geometrically generalized mesh refinement using gnns. In *International Conference on Computational Science*, pages 120–134. Springer, 2024.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

James Rowbottom, Georg Maierhofer, Teo Deveney, Eike Mueller, Alberto Paganini, Katharina Schratz, Pietro Lio, Carola-Bibiane Schönlieb, and Chris Budd. G-adaptivity: optimised graph-based mesh relocation for finite element methods. *arXiv preprint arXiv:2407.04516*, 2024.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024.

Hunar Batra, Haoqin Tu, Hardy Chen, Yuanze Lin, Cihang Xie, and Ronald Clark. Spatialthinker: Reinforcing 3d reasoning in multimodal llms via spatial rewards. *arXiv preprint arXiv:2511.07403*, 2025.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19729–19739, 2023.

Jang Hyun Cho, Boris Ivanovic, Yulong Cao, Edward Schmerling, Yue Wang, Xinshuo Weng, Boyi Li, Yurong You, Philipp Krähenbühl, Yan Wang, et al. Language-image models with 3d understanding. *arXiv preprint arXiv:2405.03685*, 2024.

Fengyun Wang, Sicheng Yu, Jiawei Wu, Jinhui Tang, Hanwang Zhang, and Qianru Sun. 3d question answering via only 2d vision-language models. *arXiv preprint arXiv:2505.22143*, 2025.

Fan Yang, Sicheng Zhao, Yanhao Zhang, Hui Chen, Haonan Lu, Jungong Han, and Guiguang Ding. Llmi3d: Mllm-based 3d perception from a single 2d image. *IEEE Transactions on Multimedia*, 2026.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. *Advances in neural information processing systems*, 37:68803–68832, 2024.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8995–9006, 2025.

C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi: 10.1002/nme.2579.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019.

Juergen Riegel, Werner Mayer, and Yorik van Havre. Freecad. *Freecadspec2002*, 2016.

OpenRouter, Inc. OpenRouter: A unified interface for LLMs. `https://openrouter.ai`, 2026. Accessed: 2026-10-11.

Olgierd C Zienkiewicz and Jian Z Zhu. A simple error estimator and adaptive procedure for practical engineerng analysis. *International journal for numerical methods in engineering*, 24(2):337–357, 1987.

A Appendix

This appendix section includes additional details of ZZ stress indicator field, our view selection module *orthoViews*, a table containing performance of various MLLMs for various load cases considered, prompt templates and demonstrations of our **GReFEM** framework in action via a few test prompts. We also release the complete CAD dataset used for our analyses.

A.1 ZZ Stress Indicator

The Cauchy stress tensor is defined via Hooke's law,

$$\boldsymbol{\sigma}(\mathbf{u}) = \mathbb{C} : \boldsymbol{\varepsilon}(\mathbf{u}), \qquad \boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\top}\right), \tag{8}$$

where $\mathbb{C}$ denotes the fourth-order elasticity tensor and $\boldsymbol{\varepsilon}(\mathbf{u})$ the infinitesimal strain tensor.

In standard finite element formulations, the stress field $\boldsymbol{\sigma}_h$ is computed element-wise and is generally discontinuous across element boundaries. The Zienkiewicz–Zhu (ZZ) error estimator addresses this limitation by constructing a smoothed or *recovered* stress field $\boldsymbol{\sigma}^*$, typically obtained via local least-squares stress recovery or patch-based averaging of neighboring element stresses Zienkiewicz and Zhu [1987, 1992].

The ZZ error indicator is defined as

$$\eta_{\mathrm{ZZ}} = \left(\int_{\Omega} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_h) : \mathbb{C}^{-1} : (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_h)\, d\Omega\right)^{1/2}, \tag{9}$$

which measures the discrepancy between the recovered and finite element stress fields in a constitutive-energy-weighted norm.

Regions with large values of η_{ZZ} are commonly interpreted as stress-critical zones where the finite element approximation is insufficiently resolved. Due to its robustness and ease of computation, the ZZ indicator is widely adopted in industrial adaptive meshing pipelines as a refinement criterion Babuška and Rheinboldt [1978].

Use as a Reference Stress Signal In our experiments, the ZZ stress indicator is computed *exclusively* on a sufficiently fine reference mesh using the corresponding reference displacement solution $\mathbf{u}_{\mathrm{ref}}$. Importantly, the ZZ field is *not* evaluated on the meshes generated by our refinement strategy, nor does it influence the refinement process itself.

Instead, the reference ZZ indicator serves as an external, physics-based validation signal, enabling us to assess whether the refinement points predicted by our method spatially align with genuinely stress-critical regions identified from the high-fidelity solution. This separation ensures that the reported stress localization results reflect true generalization rather than feedback from the refined mesh solution.

A.2 orthoViews

We introduced *orthoViews* as a view ranking module that sorts the rendered views in the order of most to least informative views considering the stress critical features like internal edges, holes and notches. We curated a dataset of 5k CAD geometries from open online sources like FreeCAD Riegel et al. [2016] and ABC Autodesk CAD dataset Koch et al. [2019].

Label Generation To generate supervision for training *orthoViews*, we first render a fixed set of candidate orthographic views for each CAD object. Views are obtained by placing a virtual camera on a sphere whose radius is scaled proportionally to the object's bounding dimensions, and systematically sweeping the camera over 12 uniformly spaced azimuth angles (0 to 360 degrees) and 11 elevation angles (-90 to 90 degrees in intervals of 18 degrees, with just one azimuth angle for the extremes), yielding a total of 110 views per mesh.

For each rendered view, we analyze the underlying CAD solid to identify stress-relevant geometric primitives, including concave internal edges, cylindrical holes, and filleted surfaces. Each view is

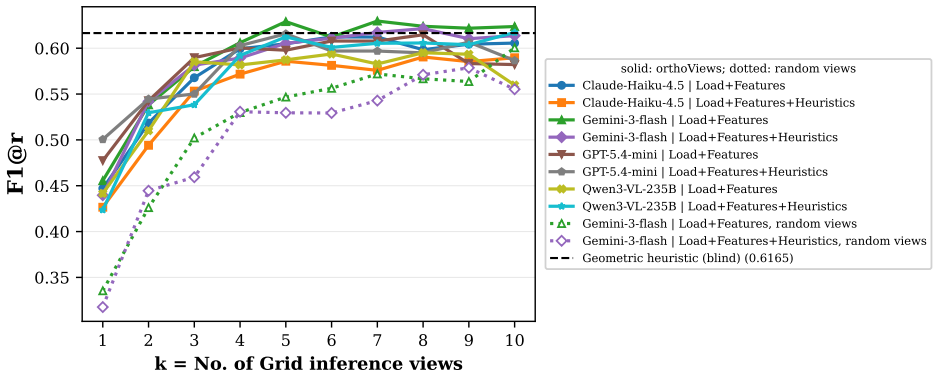


Figure 5: Anchor-level F1@r versus number of views, inference at 1–10 views. Solid lines use *orthoViews* view selection; dotted lines use randomly selected views (Gemini, both feature prompts). F1 rises steeply up to 4–5 views and plateaus thereafter. With *orthoViews*, Gemini's F1 is higher than with random views at every view count, by 0.024–0.123 (0.075–0.082 at five views). Gemini with the *Load+Features* prompt exceeds the blind geometric heuristic's F1 at five and seven or more views, and the peak F1 of every other MLLM–prompt variant lies within 0.027 of it (0.590–0.621). On the full 156-part benchmark at five views the heuristic attains higher F1@r than most MLLMs (Table 4), while **GReFEM** attains lower downstream error (Table 1).

then assigned a geometric saliency score based on the alignment of the viewing direction and the principal axes of these features, while accounting for visibility through occlusion checks on the CAD object surface. The resulting ranked view list, along with the corresponding view directions, forms the supervision signal for learning view importance in *orthoViews*. We use a cosine alignment threshold of $\tau = 0.95$ when computing feature-view alignment.

Algorithm 1 orthoViews Label Generation

Input: CAD solid $\mathcal{S}$, surface mesh $\mathcal{M}$, view directions $\mathcal{V}$
Output: Ranked views $\mathcal{V}^\star$
$\mathcal{E}_c \leftarrow \text{ExtractConcaveEdges}(\mathcal{S})$
$\mathcal{H} \leftarrow \text{ExtractCylindricalHoles}(\mathcal{S})$
$\mathcal{F} \leftarrow \text{ExtractFillets}(\mathcal{S})$
$\mathcal{G} \leftarrow \mathcal{E}_c \cup \mathcal{H} \cup \mathcal{F}$
for all $\mathbf{v} \in \mathcal{V}$ **do**
  $s[\mathbf{v}] \leftarrow 0$
  for all $g \in \mathcal{G}$ **do**
    if $\text{IsOccluded}(g, \mathbf{v}, \mathcal{M})$ **then**
      continue
    end if
    $\mathbf{a}_g \leftarrow \text{PrincipalAxis}(g)$
    if $|\mathbf{a}_g \cdot \mathbf{v}| > \tau$ **then**
      $s[\mathbf{v}] \leftarrow s[\mathbf{v}] + 1$
    end if
  end for
end for
$\mathcal{V}^\star \leftarrow \text{SortDescending}(\mathcal{V}, s)$
return $\mathcal{V}^\star$

Training The rendered views are then passed through a vision encoder, and their embeddings are aggregated and passed through a trainable self-attention layer to capture view ranking. We employ a pretrained frozen DINOv2 encoder as it captures both global semantics (e.g., object-level understanding) and local details (e.g., object parts, textures, edges), making it ideal for representing geometric features of objects visible in the views.

Table 4: **Anchor-level localization** on the expanded benchmark (156 parts × 5 loads, top five *orthoViews* views). Precision@r, Recall@r and F1@r against the top-1% reference ZZ points, $r = 4h_{\min}$, averaged over cases; n is the number of cases in which the variant returned anchors. Avg Pts: mean anchors per case. Cell mentions/view: grid-cell indices returned per view, summed over feature categories (a cell listed under several categories counts several times). The blind heuristic is reported at its native anchor count and reduced to **GReFEM**-Gemini's (*Load+Features+Heuristics*) count per case. Localization overlap does not reproduce the downstream ranking of Table 1 (Sec. 5.2).

Model / Baseline	Prompt Type	P@r	R@r	F1@r	Avg Pts (k)	Cell mentions/view	n
Gemini-3-flash-preview	Load+Features+Heuristics	0.553	0.619	0.523	4.7	18.6	756
	Load+Features	0.552	0.651	0.540	4.9	22.1	758
	Load-Only	0.539	0.692	0.547	5.4	9.6	764
Claude-Haiku-4.5	Load+Features+Heuristics	0.545	0.561	0.485	4.2	15.3	748
	Load+Features	0.537	0.622	0.517	4.8	19.9	767
	Load-Only	0.525	0.721	0.551	5.9	15.2	774
GPT-5.4-mini	Load+Features+Heuristics	0.549	0.644	0.526	4.4	38.6	769
	Load+Features	0.547	0.643	0.526	4.3	36.6	757
	Load-Only	0.520	0.778	0.579	6.5	25.5	771
Qwen3-VL-235B	Load+Features+Heuristics	0.572	0.577	0.504	3.4	62.9	735
	Load+Features	0.565	0.585	0.505	3.8	50.3	717
	Load-Only	0.534	0.648	0.522	4.5	18.2	735
Geometric heuristic (blind)	native count	0.567	0.822	0.609	6.0	–	780
	GReFEM-Gemini count	0.571	0.770	0.596	3.7	–	756
Load-informed heuristic	native count	0.601	0.488	0.469	1.9	–	780

Formally, given V **GReFEM** views $\{\mathcal{I}_k\}_{k=1}^{V}$, a frozen DINOv2 encoder f_θ maps each view to an embedding

$$\mathbf{e}_k = f_\theta(\mathcal{I}_k) \in \mathbb{R}^D. \tag{10}$$

The embeddings are stacked as $\mathbf{e} \in \mathbb{R}^{V \times D}$ and passed through a self-attention module

$$\tilde{\mathbf{Z}} = \mathrm{Attn}(\mathbf{e}, \mathbf{e}, \mathbf{e}), \tag{11}$$

followed by an MLP head g_ϕ to predict scalar view scores $s_v = g_\phi(\tilde{\mathbf{z}}_v)$. The self attention layer and the scoring MLP are both kept trainable.

The model is trained using a pairwise ranking loss

$$\mathcal{L} = \mathbb{E}_{(v^+, v^-)}\big[\max(0, m - (s_{v^+} - s_{v^-}))\big], \tag{12}$$

s_{v^+} and s_{v^-} denote the geometrically preferred and non-preferred scores, respectively, constructed from oracle view scores in Algorithm 1, m is a margin hyperparameter enforcing a minimum score separation, kept here equal to 1.

Table 7: **Element-level localization** on the final meshes over all 780 cases, against the top-0.1% reference ZZ hotspots. "Predictions" are the deeply refined elements (size $\leq 1.25\,h_{\min}$): element precision **P** is the fraction of these within r of a hotspot, element recall **R** the fraction of hotspots covered. Element size at hotspots is the median edge-length proxy (cube root of tetrahedron volume) of elements within r of a hotspot. Element-level F1 is nearly identical across the three policies, but the element size at the hotspots follows the downstream ranking: paired per case, **GReFEM**'s hotspot elements are smaller than the blind heuristic's in 66% of cases and larger than the load-informed heuristic's in 68%.

method	P	R	F1	elem. size at hotspots (med.)	δ_{vM}^{p99} mean
GReFEM	0.318	0.903	0.432	0.45	0.626
Geometric heuristic (blind)	0.313	0.922	0.427	0.49	0.672
Load-informed heuristic	0.350	0.829	0.432	0.42	0.567

Table 5: **Downstream FEM error per load case** at a matched element budget, all 156 parts (mean of δ_{vM}^{p99} / δ_E^{crit}; lower is better for both). **GReFEM** rows use the *Load+Features+Heuristics* prompt. Last block: paired median difference in δ_{vM}^{p99}, **GReFEM** (Gemini) minus baseline (negative favours **GReFEM**), with two-sided Wilcoxon p-values.

Policy	Compression	Bending	Torsion	Bending + Comp.	Torsion + Comp.
Coarse (no refinement)	0.739/0.948	0.733/0.940	0.730/0.943	0.742/0.941	0.744/0.943
GReFEM (Gemini)	0.659/0.899	0.623/0.869	0.622/0.868	0.600/0.850	0.617/0.896
GReFEM (GPT-5.4-mini)	0.655/0.889	0.637/0.880	0.623/0.871	0.607/0.851	0.624/0.863
GReFEM (Claude-Haiku-4.5)	0.671/0.897	0.654/0.889	0.639/0.862	0.608/0.858	0.637/0.879
GReFEM (Qwen3-VL-235B)	0.673/0.901	0.660/0.885	0.643/0.878	0.626/0.859	0.642/0.894
Geometric heuristic (blind)	0.698/0.911	0.697/0.905	0.658/0.877	0.645/0.860	0.656/0.892
Load-informed heuristic	0.633/0.888	0.574/0.872	0.514/0.851	0.550/0.831	0.562/0.888
ZZ from coarse solve	0.710/0.919	0.688/0.898	0.652/0.879	0.630/0.838	0.671/0.877
AMBER (learned AMR)	0.616/0.860	0.654/0.893	0.638/0.864	0.610/0.867	0.602/0.859
ZZ from fine reference solve	*0.607/0.858*	*0.572/0.874*	*0.570/0.859*	*0.472/0.791*	*0.523/0.857*
*Paired $\Delta\delta_{vM}^{p99}$, **GReFEM** (Gemini) − baseline (median, p)*					
vs. geometric heuristic	-0.021 (0.013)	-0.064 ($5{\cdot}10^{-7}$)	-0.027 (0.020)	-0.030 (0.007)	-0.005 (0.038)
vs. load-informed heuristic	$+0.032$ (0.034)	$+0.013$ (0.042)	$+0.065$ ($8{\cdot}10^{-6}$)	$+0.027$ (0.049)	$+0.042$ (0.004)
vs. ZZ from coarse solve	-0.040 (0.001)	-0.040 ($5{\cdot}10^{-5}$)	-0.039 (0.005)	-0.030 (0.082)	-0.047 (0.001)
vs. AMBER	$+0.053$ (0.035)	-0.017 (0.15)	-0.022 (0.50)	-0.031 (0.53)	$+0.029$ (0.15)

Table 6: **Sensitivity of GReFEM's downstream error across experimental dimensions** (all 156 parts). For each dimension, the mean relative error is computed per type, pooling over all other dimensions at $h \times 1.0$ (for the sizing level, pooling over MLLMs and prompts at each type), and the standard deviation and range across levels are reported. Prompt types use native element counts. The last row gives, for reference, the spread across the five attainable refinement policies (GReFEM-Gemini, both heuristics, coarse-solve ZZ, AMBER).

Dimension	Types	δ_{vM}^{p99} std	δ_{vM}^{p99} range	δ_E^{crit} std	δ_E^{crit} range
MLLM	4	0.006	0.018	0.005	0.014
Prompt	3	0.002	0.005	0.005	0.011
Load case	5	0.021	0.063	0.014	0.041
Sizing level	4	0.044	0.116	0.024	0.064
Refinement policy (reference)	5	0.039	0.105	0.009	0.023

A.3 Expanded Downstream Evaluation

This appendix expands the matched-budget downstream study of the main text over the full 156-part ($\times$ 5 load) benchmark. All errors are relative to the $\approx 5 \times 10^6$-element reference solve and measured inside the critical region Ω_{crit}.

Load-informed heuristic. This policy keeps the *same* geometric candidates as the blind heuristic and changes only *which* of them are refined, using a solver-free estimate of nominal stress. We voxel-fill the part and slice it into thin layers along the loading axis y (clamped at $y_{\min}$, loaded at $y_{\max}$); for each layer we compute purely geometric section properties, area $A(y)$, centroid $(\bar{x}, \bar{z})$, second moment of area $I_z(y)$, and polar moment $J(y)$. Each candidate is assigned to its nearest layer and weighted by the textbook nominal stress for the load case: $w = 1/A$ (compression, net-section stress), $w = |x - \bar{x}|/I_z$ (bending, $\sigma = Mc/I$), and $w = r/J$ (torsion, $\tau = Tr/J$), with r the distance to the layer centroid; combined loads sum the normalised terms. We keep the top stress-weighted 30% of candidates and density-filter to the matched anchor budget. Geometry thus decides *where* refinement may go and beam-theory stress decides *which* locations matter; this prior wins where closed-form reasoning is valid but must be re-derived by hand for each load family.

Learned-AMR baseline (AMBER). We retrained AMBER Freymuth et al. [2025], an iterative graph network that predicts a continuous sizing field, from scratch on this task: 440 expert adaptive meshes (88 ABC training parts $\times$ 5 load cases, disjoint from all 156 evaluation parts), each expert mesh built by projecting the fine reference solve's top-percentile ZZ refinement points onto the coarse-resolution mesh, trained to its validation plateau on H100 GPUs (our 440 expert meshes are

4–$20\times$ more than the original AMBER datasets), with load-case conditioning added. At test time its sizing field is globally scaled to **GReFEM**'s element budget at each sweep level; it fails to mesh $\approx 6\%$ of cases. The final-step scale is calibrated by a secant search (up to 10 iterations, 5% tolerance), which brings 95% of AMBER's meshes within 10% of **GReFEM**'s element count at every sizing level; the remainder are geometries on which the predicted field cannot be meshed. Across the sizing sweep (Table 2) the two policies are comparable: their mean errors differ by at most 0.032 in peak stress and 0.012 in strain energy at every level. A zero-shot, prompt-based policy thus lands in the same accuracy range as a sizing field supervised on 440 expert meshes; on a narrow part family with abundant expert meshes, a supervised sizing field would likely pull further ahead.

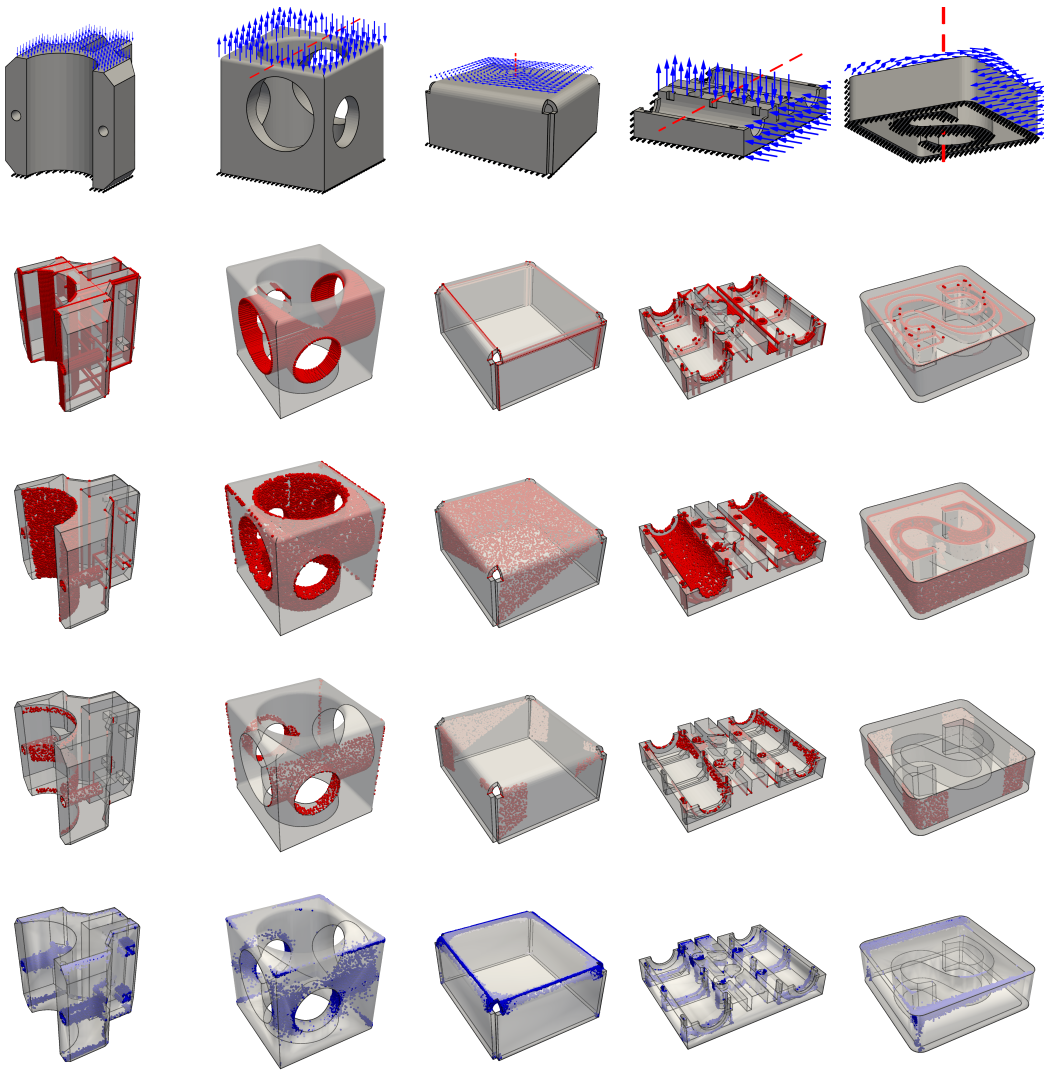


Figure 6: Qualitative visual comparison across experimental loading cases (row 1), predicted refinement points via **GReFEM** (row 2), blind geometric heuristic points (row 3), load-informed heuristic points (row 4), and ground-truth ZZ stress indicator fields (row 5). All anchor rows show the native anchor sets before matching to **GReFEM**'s anchor count. The columns represent different object geometries and loading scenarios. All results were generated using a maximum of five **GReFEM** views selected via the *orthoViews* module using **GReFEM**-gemini (*Load+Features*). As is apparent from the figures, the blind geometric heuristic spreads refinement points over all concave edges and surfaces that could be stress critical, whereas **GReFEM** concentrates them on the features activated by the given load, and the load-informed heuristic keeps the subset of the blind candidates with the highest closed-form nominal stress. Quantitatively (Sec. 5.1), this selection lowers critical-region peak-stress error at a matched budget, even though anchor-level overlap with the reference ZZ field is not higher than the heuristic's.

A.4 Prompts

We prompt the MLLMs in **GReFEM** to return region proposals from gridded view images. We introduce the default prompt template for the two physics informed prompt types introduced in the paper with place holders for perspective views, gridded views and loading information followed by example outputs for each of the model compared in our study. For the *Load+Features+Heuristics* prompt template we append "{additional_info}" in the prompt, while for *Load+Features* prompt we omit it, keeping all else same.

Region Localization Prompt Template - *Load+Features+Heuristics/Load+Features*

System Prompt:

```
You are an expert in computational geometry and finite element meshing.
```

User Prompt:

```
You are given multiple images of a single CAD part along with their
camera angles marked in (e: elevation, a: azimuth) format.
These angles symbolize the location of the camera with respect to the
objects' center in spherical coordinates.
The first {num_perspective_views} images show general 3D views of the
object (not gridded) with camera situated at various angles respectively:
{angles_list_perspective_views}.
The remaining {num_gridded_views} images show orthographic views with
camera positions in the list respectively
(e: elevation, a: azimuth): {angles_list_gridded_views}.
Each orthographic view has a visible grid with numbered cells.
{intro_loading[loading_case]}
The object is completely fixed on surfaces as indicated by the
Protruding dashed black lines
Task:
For EACH gridded orthographic image, identify all grid cells that
contain inside them these features that will be stress critical for the
given loading condition:
1) Internal corner edges (concave edges that fold into the object and go
into the view plane).
2) Extruded Contour/Protruding Contour (extruded closed contour cut
partially like a blind hole into the object or protruding out of the
object like a boss)
3) Through Holes (any internal surface with hollow space inside all the
way).
4) Concave Fillet Surfaces, or arches (Any concave curved surface)
I.C.E: Internal Corner Edge, E.C/P.C: Extruded Contour/Protruding
Contour, T.H: Through Holes, F: Concave Fillet Surfaces or arches
Use the non-gridded {num_perspective_views} 3D views ONLY to understand
the overall shape. The grid numbers should be marked only for the
corresponding gridded orthographic images.
Base all cell predictions ONLY on what is visible in the corresponding
gridded orthographic image.
For Hollow features mark the boundary cells that contain the hollow
feature inside them.
Do NOT guess or infer cells that are not clearly visible in that image.
Output format (repeat for each gridded image):
***I.C.E: c1, c2, c3 ***
***E.C/P.C: c4, c5 ***
***T.H: c6, c7 ***
***F: c8, c9 ***
{additional_info[loading_case]}
Rules:
```

```
- List only cell numbers.
- If no cells apply, write: NONE.
- DO NOT Add explanations
```

Region Localization Prompt Template - Load-Only

System Prompt:

```
You are an expert in computational geometry and finite element meshing.
```

User Prompt:

```
You are given multiple images of a single CAD part along with their
camera angles marked in (e: elevation, a: azimuth) format. These angles
symbolize the location of the camera with respect to the objects'
center in spherical coordinates.
The first {num_perspective_views} images show general 3D views of
the object (not gridded) with camera situated at various angles
respectively: {angles_list_perspective_views}.
The remaining {num_gridded_views} images show orthographic views with
camera positions in the list respectively (e: elevation, a: azimuth):
{angles_list_gridded_views}.
Each orthographic view has a visible grid with numbered cells.
{intro_loading[loading_case]}
Task:
For EACH gridded orthographic image, identify all grid cells that
contain inside them stress critical features.
Use the non-gridded {num_perspective_views} 3D views ONLY to understand
the overall shape. The grid numbers should be marked only for the
corresponding gridded orthographic images.
Base all cell predictions ONLY on what is visible in the corresponding
gridded orthographic image.
Do NOT guess or infer cells that are not clearly visible in that image.
Output format (repeat for each gridded image):
***Cells: c1, c2, c3 ***
Rules:
- List only cell numbers.
- If no cells apply, write: NONE.
- DO NOT Add explanations
```

Loading Case Specific Information

Loading Introduction:

```
Compression: A compressive displacement load acts on the object in
the -y direction on its top surface as indicated by the blue arrows.

Bending: A bending displacement load acts  on the object in the +y/-y
direction on the top surface as indicated by the blue arrows about the
horizontal axis indicated via dashed red line.

Torsion: A torsion displacement load acts on the object in the
anti-clockwise direction along the x-z plane as indicated by the blue
curved arrows about the vertical axis of loading passing through the
center of the top surface as indicated by the dashed red line.

Bending-Compression: A bending displacement load acts on the object in
the +y/-y direction on the top surface as indicated by the blue arrows
about the horizontal axis indicated via dashed red line. Also a
```

compressive load acts on the object in the +/- x direction as shown by the blue arrows.

Torsion-Compression: A torsion displacement load acts on the object in the anti-clockwise direction along the x-z plane as indicated by the blue curved arrows about the vertical axis of loading passing through the center of the top surface as indicated by the dashed red line. Also the compressive load acts on the object in the +/- x direction as shown by the blue arrows.

Additional Information for Spotting Stress Critical Features:

Compression: For the top and bottom views, only consider the BH features. All edges or surfaces along the loading direction are not stress critical as they are not perpendicular to the load and do not create stress concentrations. Neglect the features too far away if the load acts on a narrow part of the top surface passing through the center of the object as they are not stress critical.

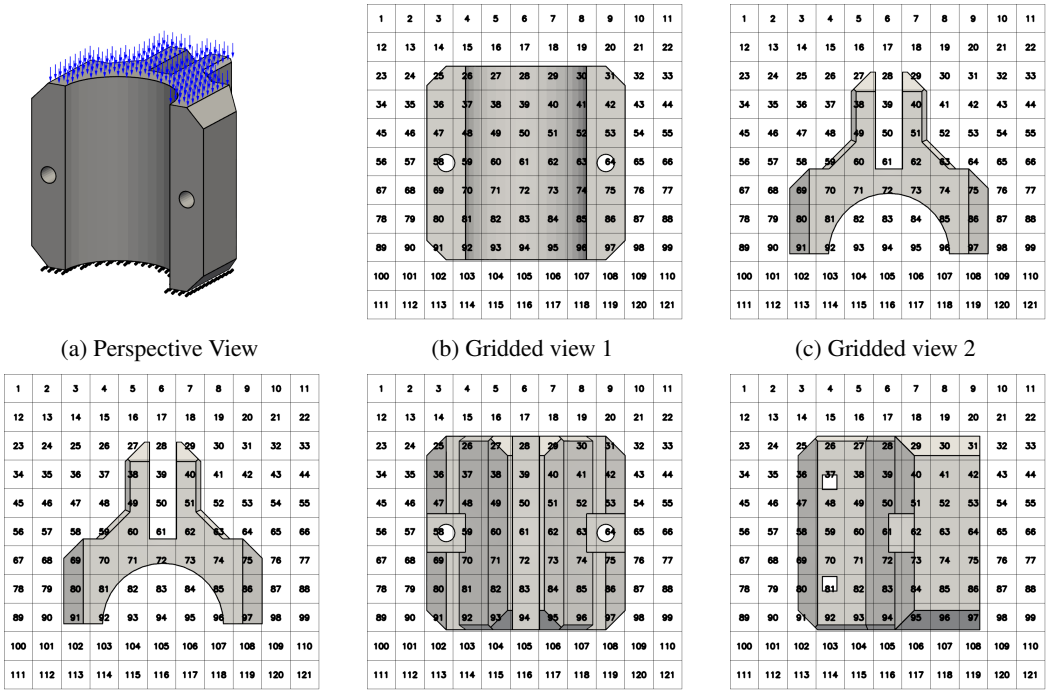
Bending: For the top and bottom views, only consider the BH features. All edges or surfaces along the loading direction are not stress critical as they are not perpendicular to the load and do not create stress concentrations. Geometric features ICE, BH, TH too close to the axis about which load is applied are usually not stress critical, in the case of bending loads features away from the axis are more likely to be stress critical.

Torsion: Geometric features ICE, BH, TH too close to the axis about which load is applied are usually not stress critical, in the case of torsion loads features away from the axis are more likely to be stress critical.

Bending-Compression: For each load (bending or compression), only consider the BH features when viewed parallel to the loading direction. All edges or surfaces along the loading direction for each load are not stress critical but perpendicular to the respective loads are. Geometric features ICE, BH, TH too close to the axis about which bending load is applied are usually not stress critical, in the case of bending loads features away from the axis are more likely to be stress critical. Consider stress critical areas taking care of both the loads.

Torsion-compression: For the torsion load geometric features ICE, BH, TH too close to the axis about which load is applied are usually not stress critical, in the case of torsion loads features away from the axis are more likely to be stress critical. Also, features along the loading direction of the compressive load are not stress critical as they do not create stress concentrations. Consider stress critical areas taking care of both the loads.

Using the prompts described above, we query each of the MLLMs considered in the previous section and report their responses. For each query, we provide a perspective views of the geometry and the top 5 orthographic views selected by *orthoViews*.



(a) Perspective View (b) Gridded view 1 (c) Gridded view 2

(d) Gridded view 3 (e) Gridded view 4 (f) Gridded view 5

Figure 7: Five images provided jointly with example prompt 1. Top row: non-gridded perspective views for global 3D context. Bottom row: gridded orthographic views used for cell-level region localization.

A.4.1 Example Prompt 1: Compression (*Load+Features+Heuristics*, views=5, Figure 7)

Response: Gemini 3 Flash Preview

```
 Gridded Image ('e0', 'a0'):
***I.C.E: 26, 30, 37, 41, 48, 52, 59, 63, 70, 74, 81, 85, 92, 96***
***E.C/P.C: NONE***
***T.H: 58, 64***
***F: NONE***

Gridded Image ('e90', 'a0'):
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: NONE***
***F: NONE***

Gridded Image ('e-90', 'a0'):
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: NONE***
***F: 71, 72, 73, 82, 83, 84***

Gridded Image ('e0', 'a180'):
***I.C.E: 28, 39, 50, 61, 72, 83, 94***
***E.C/P.C: 58, 59, 63, 64***
***T.H: 58, 64***
***F: NONE***
```

```
Gridded Image ('e0', 'a270'):
***I.C.E: 28, 39, 50, 61, 72, 83, 94***
***E.C/P.C: 61, 62, 72, 73***
***T.H: 37, 81***
***F: NONE***
```

Response: Claude Haiku 4.5

```
Response: Front View (e0, a0)

***I.C.E: 26, 27, 28, 29, 30, 31, 37, 38, 39, 40, 41, 42, 48, 49, 50,
51, 52, 53, 59, 60, 61, 62, 63, 64***

***E.C/P.C: 37, 38, 39, 40, 41, 42, 59, 60, 61, 62, 63, 64***

***T.H: NONE***

***F: 82, 83, 84, 85***

## Image 2: Top View (e90, a0)

***I.C.E: NONE***

***E.C/P.C: NONE***

***T.H: NONE***

***F: NONE***

## Image 3: Bottom View (e-90, a0)

***I.C.E: NONE***

***E.C/P.C: NONE***

***T.H: NONE***

***F: NONE***

## Image 4: Back View (e0, a180)

***I.C.E: 26, 27, 28, 29, 30, 31, 37, 38, 39, 40, 41, 42, 48, 49, 50,
51, 52, 53, 59, 60, 61, 62, 63, 64***

***E.C/P.C: 37, 38, 39, 40, 41, 42, 59, 60, 61, 62, 63, 64***

***T.H: NONE***

***F: 82, 83, 84, 85***

## Image 5: Right View (e0, a270)

***I.C.E: 26, 27, 28, 29, 30, 31, 37, 38, 39, 40, 41, 42, 48, 49, 50,
51, 52, 53, 59, 60, 61, 62, 63, 64***

***E.C/P.C: 37, 38, 39, 40, 41, 42, 59, 60, 61, 62, 63, 64***
```

```
***T.H: 37, 38, 39, 40, 41, 42, 59, 60, 61, 62, 63, 64***

***F: 82, 83, 84, 85***
```

Response: GPT 5.4 mini

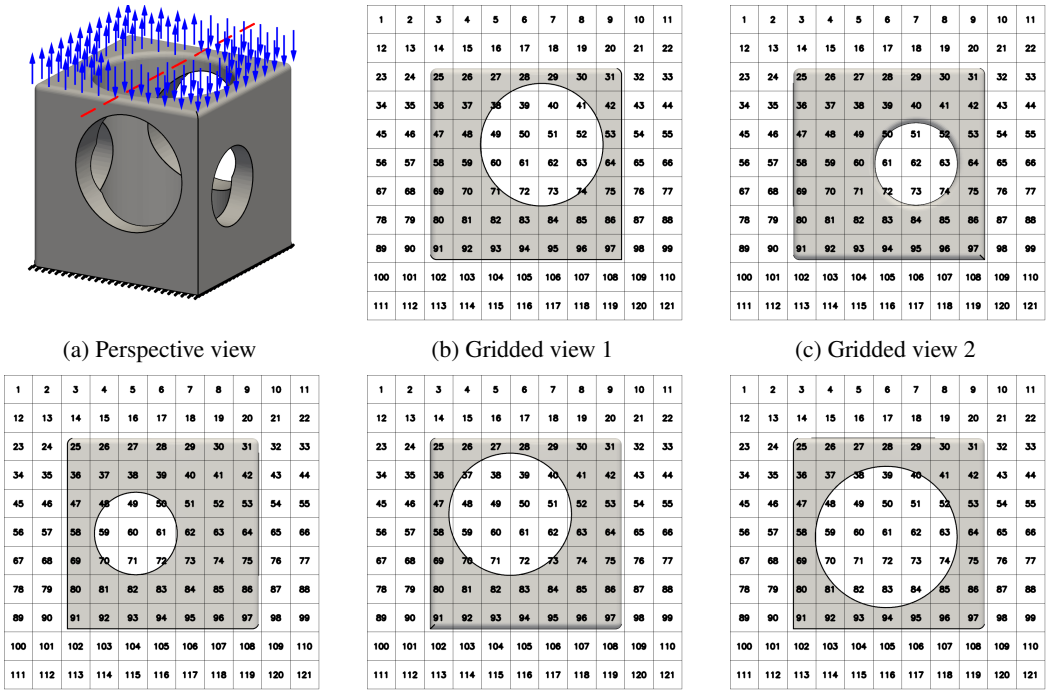
```
***I.C.E: 27, 28, 29, 38, 39, 40, 49, 50, 51, 59, 60, 61, 62, 63, 71,
72, 73, 74, 75, 81, 82, 83, 84, 85, 86, 91, 92, 93, 94, 95, 96,
97 ***
***E.C/P.C: 26, 30, 37, 41, 48, 52, 58, 64, 70, 74, 80, 86, 92, 96 ***
***T.H: NONE ***
***F: 27, 29, 31, 38, 40, 42, 49, 51, 53, 59, 63, 69, 75, 81, 87, 91,
97 ***

***I.C.E: 27, 28, 29, 38, 39, 40, 49, 50, 51, 59, 60, 61, 62, 63, 71,
72, 73, 74, 75, 81, 82, 83, 84, 85, 86, 91, 92, 93, 94, 95, 96, 97 ***
***E.C/P.C: 26, 30, 37, 41, 48, 52, 58, 64, 70, 74, 80, 86, 92, 96 ***
***T.H: NONE ***
***F: 27, 29, 31, 38, 40, 42, 49, 51, 53, 59, 63, 69, 75, 81, 87, 91,
97 ***

***I.C.E: 26, 27, 28, 29, 38, 39, 40, 41, 49, 50, 51, 52, 60, 61, 62,
63, 71, 72, 73, 74, 82, 83, 84, 85, 92, 93, 94, 95 ***
***E.C/P.C: 25, 30, 36, 42, 47, 53, 58, 64, 69, 75, 91, 97 ***
***T.H: 58, 64 ***
***F: 25, 27, 29, 38, 40, 49, 51, 59, 63, 69, 75, 91, 97 ***

***I.C.E: 25, 26, 27, 28, 29, 37, 38, 39, 40, 41, 47, 48, 49, 50, 51,
58, 59, 60, 61, 62, 69, 70, 71, 72, 73, 80, 81, 82, 83, 84, 91, 92, 93,
94, 95 ***
***E.C/P.C: 25, 30, 36, 42, 47, 53, 58, 64, 69, 75, 91, 97 ***
***T.H: 58, 64 ***
***F: 25, 27, 29, 37, 40, 42, 47, 49, 51, 58, 62, 69, 73, 80, 84, 91,
95 ***

***I.C.E: 25, 27, 29, 36, 38, 40, 47, 49, 51, 58, 60, 62, 69, 71, 73,
80, 82, 84, 91, 93, 95 ***
***E.C/P.C: 26, 30, 37, 41, 48, 52, 59, 63, 70, 74, 92, 96 ***
***T.H: 37, 81 ***
***F: 25, 29, 40, 47, 51, 58, 62, 69, 73, 80, 84, 91, 95 ***
```

Response: Qwen3 VL 235B A22B Instruct

```
***I.C.E: 58, 64 ***
***E.C/P.C: 25, 31, 91, 97 ***
***T.H: NONE ***
***F: NONE ***

***I.C.E: 49, 63 ***
***E.C/P.C: 27, 29, 69, 75, 91, 97 ***
***T.H: NONE ***
***F: 82, 83, 84, 85, 92, 93, 94, 95, 96 ***

***I.C.E: 49, 63 ***
***E.C/P.C: 27, 29, 69, 75, 91, 97 ***
***T.H: NONE ***
```

```
***F: 82, 83, 84, 85, 92, 93, 94, 95, 96 ***

***I.C.E: 58, 64 ***
***E.C/P.C: 25, 31, 91, 97 ***
***T.H: NONE ***
***F: NONE ***

***I.C.E: 37, 81 ***
***E.C/P.C: 26, 28, 36, 38, 47, 49, 50, 51, 52, 61, 62, 63, 64, 72, 73,
74, 75, 80, 82, 83, 84, 85, 86, 91, 92, 93, 94, 95, 96, 97 ***
***T.H: NONE ***
***F: NONE ***
```

A.4.2 Example Prompt 2: Bending (*Load+Features*, views=5, Figure 8)



(a) Perspective view (b) Gridded view 1 (c) Gridded view 2

(d) Gridded view 3 (e) Gridded view 4 (f) Gridded view 5

Figure 8: Five images provided jointly with example prompt 2. Top row: non-gridded perspective views for global 3D context. Bottom row: gridded orthographic views used for cell-level region localization.

Response: Gemini 3 Flash Preview

```
 Image 3: (e0, a0)
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: 28, 29, 38, 39, 40, 41, 49, 50, 51, 52, 60, 61, 62, 63, 71, 72,
73, 74, 83, 84***
***F: NONE***

Image 4: (e0, a270)
***I.C.E: NONE***
```

```
***E.C/P.C: NONE***
***T.H: 50, 51, 52, 61, 62, 63, 72, 73, 74***
***F: NONE***

Image 5: (e0, a90)
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: 48, 49, 50, 59, 60, 61, 70, 71, 72***
***F: NONE***

Image 6: (e0, a180)
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: 27, 28, 29, 30, 37, 38, 39, 40, 41, 48, 49, 50, 51, 52, 59, 60,
61, 62, 63, 70, 71, 72, 73, 74, 82, 83, 84, 85***
***F: NONE***

Image 7: (e-90, a0)
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: 38, 39, 40, 41, 48, 49, 50, 51, 52, 59, 60, 61, 62, 63, 70, 71,
72, 73, 74, 82, 83, 84, 85***
***F: NONE***
```

Response: Claude Haiku 4.5

```
Response: (e0, a0) - Front View

***I.C.E: 48, 49, 50, 51, 52, 59, 60, 61, 62, 63***
***E.C/P.C: NONE***
***T.H: 50, 51, 52, 61, 62, 63***
***F: 48, 49, 50, 51, 52, 59, 60, 61, 62, 63***

## Image 2: (e0, a270) - Left View

***I.C.E: 25, 26, 31, 32, 36, 37, 42, 43, 47, 48, 52, 53, 58, 59, 63,
64, 69, 70, 74, 75, 80, 81, 85, 86, 91, 92, 96, 97***
***E.C/P.C: NONE***
***T.H: NONE***
***F: NONE***

## Image 3: (e0, a90) - Right View

***I.C.E: 25, 26, 31, 32, 36, 37, 42, 43, 47, 48, 52, 53, 58, 59, 63,
64, 69, 70, 74, 75, 80, 81, 85, 86, 91, 92, 96, 97***
***E.C/P.C: NONE***
***T.H: NONE***
***F: NONE***

## Image 4: (e0, a180) - Back View

***I.C.E: 48, 49, 50, 51, 52, 59, 60, 61, 62, 63***
***E.C/P.C: NONE***
***T.H: 50, 51, 52, 61, 62, 63***
***F: 48, 49, 50, 51, 52, 59, 60, 61, 62, 63***

## Image 5: (e-90, a0) - Bottom View
```

```
***I.C.E: 25, 26, 31, 32, 36, 37, 42, 43, 47, 48, 52, 53, 58, 59, 63,
64, 69, 70, 74, 75, 80, 81, 85, 86, 91, 92, 96, 97***
***E.C/P.C: NONE***
***T.H: 50, 51, 52, 61, 62, 63***
***F: NONE***
```

Response: GPT 5.4 mini

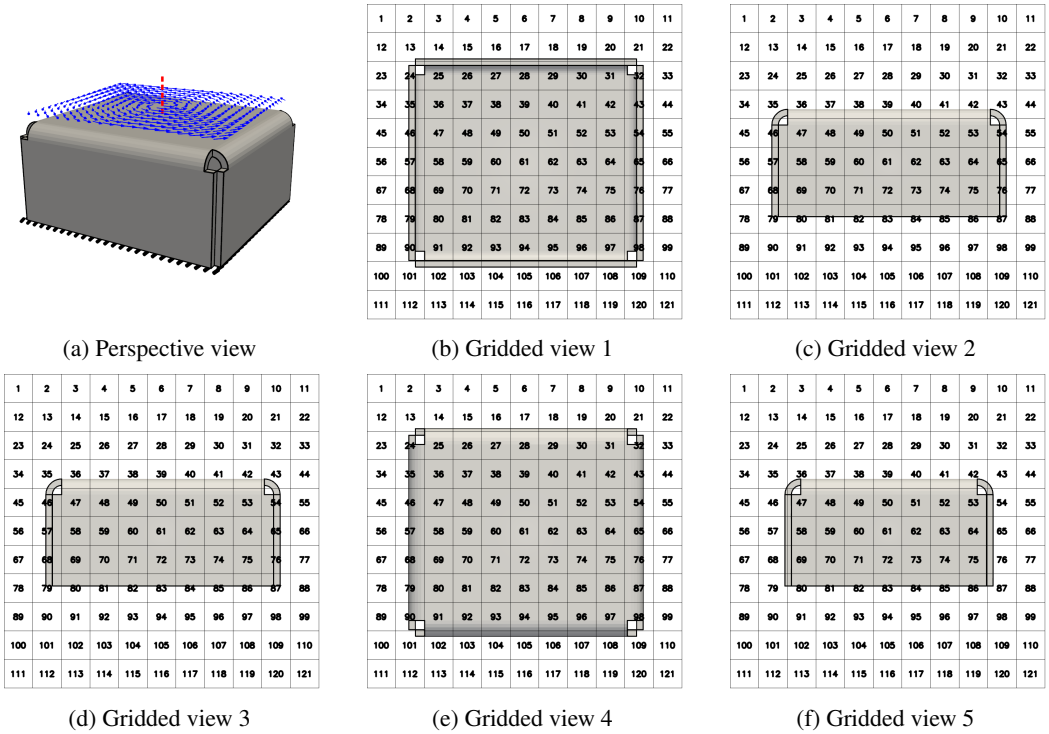
```
***I.C.E: 25, 31, 91, 97 ***
***E.C/P.C: NONE ***
***T.H: 36, 37, 38, 39, 40, 41, 42, 47, 48, 49, 50, 51, 52, 53, 58, 59,
60, 61, 62, 63, 64, 69, 70, 71, 72, 73, 74, 75, 80, 81, 82, 83, 84, 85,
86, 91, 92, 93, 94, 95, 96, 97 ***
***F: 25, 26, 27, 28, 29, 30, 31, 36, 42, 47, 53, 58, 64, 69, 75, 80,
86, 91, 92, 93, 94, 95, 96, 97 ***

***I.C.E: 25, 31, 91, 97 ***
***E.C/P.C: NONE ***
***T.H: 25, 26, 27, 28, 29, 30, 31, 36, 37, 38, 39, 40, 41, 42, 47, 48,
49, 50, 51, 52, 53, 58, 59, 60, 61, 62, 63, 64, 69, 70, 71, 72, 73, 74,
75, 80, 81, 82, 83, 84, 85, 86, 91, 92, 93, 94, 95, 96, 97 ***
***F: 25, 26, 27, 28, 29, 30, 31, 37, 38, 39, 40, 41, 42, 48, 49, 50,
51, 52, 53, 59, 60, 61, 62, 63, 64, 70, 71, 72, 73, 74, 75, 81, 82, 83,
84, 85, 86, 91, 92, 93, 94, 95, 96, 97 ***

***I.C.E: 25, 91 ***
***E.C/P.C: NONE ***
***T.H: 38, 39, 40, 41, 48, 49, 50, 51, 52, 59, 60, 61, 62, 63, 70, 71,
72, 73, 74, 81, 82, 83, 84, 85, 92, 93, 94, 95, 96 ***
***F: 25, 26, 27, 28, 29, 30, 31, 38, 39, 40, 41, 48, 49, 50, 51, 52,
59, 60, 61, 62, 63, 70, 71, 72, 73, 74, 81, 82, 83, 84, 85, 92, 93, 94,
95, 96 ***

***I.C.E: 25, 31, 91, 97 ***
***E.C/P.C: NONE ***
***T.H: 38, 39, 40, 41, 48, 49, 50, 51, 52, 59, 60, 61, 62, 63, 70, 71,
72, 73, 74, 81, 82, 83, 84, 85, 92, 93, 94, 95, 96 ***
***F: 25, 26, 27, 28, 29, 30, 31, 38, 39, 40, 41, 48, 49, 50, 51, 52,
59, 60, 61, 62, 63, 70, 71, 72, 73, 74, 81, 82, 83, 84, 85, 92, 93, 94,
95, 96 ***

***I.C.E: 25, 31, 91, 97 ***
***E.C/P.C: NONE ***
***T.H: 25, 26, 27, 28, 29, 30, 31, 36, 37, 38, 39, 40, 41, 42, 47, 48,
49, 50, 51, 52, 53, 58, 59, 60, 61, 62, 63, 64, 69, 70, 71, 72, 73, 74,
75, 80, 81, 82, 83, 84, 85, 86, 91, 92, 93, 94, 95, 96, 97 ***
***F: 25, 26, 27, 28, 29, 30, 31, 36, 42, 47, 53, 58, 64, 69, 75, 80,
86, 91, 92, 93, 94, 95, 96, 97 ***
```

Response: Qwen3 VL 235B A22B Instruct

```
***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: 38,39,40,41,42,48,49,50,51,52,58,59,60,61,62,68,69,70,71,72,
78,79,80,81,82 ***
***F: NONE ***
```

```
***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: 50,51,52,53,61,62,63,64,72,73,74,75,83,84,85,86,94,95,96,97 ***
***F: NONE ***

***I.C.E: NONE ***
***E.C.P.C: NONE ***
***T.H: 48,49,50,51,52,58,59,60,61,62,68,69,70,71,72,78,79,80,81,82 ***
***F: NONE ***

***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: 37,38,39,40,41,47,48,49,50,51,57,58,59,60,61,67,68,69,70,71,77,
78,79,80,81 ***
***F: NONE ***

***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: 38,39,40,41,42,48,49,50,51,52,58,59,60,61,62,68,69,70,71,72,
78,79,80,81,82 ***
***F: NONE ***
```

A.4.3 Example Prompt 3: Torsion (Load-Only, views=5, Figure 9)



(a) Perspective view (b) Gridded view 1 (c) Gridded view 2

(d) Gridded view 3 (e) Gridded view 4 (f) Gridded view 5

Figure 9: Five images provided jointly with example prompt 3. Top row: non-gridded perspective views for global 3D context. Bottom row: gridded orthographic views used for cell-level region localization.

Response: Gemini 3 Flash Preview

```
Image 1 [('e-90', 'a0')]:
***Cells: 24, 32, 90, 98***

Image 2 [('e0', 'a0')]:
***Cells: 46, 54***

Image 3 [('e0', 'a180')]:
***Cells: 46, 54***

Image 4 [('e90', 'a0')]:
***Cells: 24, 32, 90, 98***

Image 5 [('e0', 'a270')]:
***Cells: 47, 53***
```

Response: Claude Haiku 4.5

```
Response: I'll analyze each gridded orthographic image to identify cells
containing stress critical features under torsional loading about the
vertical axis.

**Image 1 (e-90, a0) - Bottom view:**
The visible stress critical features are the corner cutouts/notches at
the four corners of the rectangular part.

***Cells: 24, 32, 101, 109***

**Image 2 (e0, a0) - Front view:**
The visible stress critical features are the corner cutouts/notches on
the left and right sides.

***Cells: 24, 32, 101, 109***

**Image 3 (e0, a180) - Back view:**
The visible stress critical features are the corner cutouts/notches on
the left and right sides (same as front view, mirrored).

***Cells: 24, 32, 101, 109***

**Image 4 (e90, a0) - Top view:**
The visible stress critical features are the corner cutouts/notches at
the four corners and the slot features along the edges.

***Cells: 24, 25, 32, 101, 102, 109***

**Image 5 (e0, a270) - Right side view:**
The visible stress critical features are the corner cutouts/notches on
the top and bottom.

***Cells: 36, 37, 46, 47, 80, 81, 90, 91***
```

Response: GPT 5.4 mini

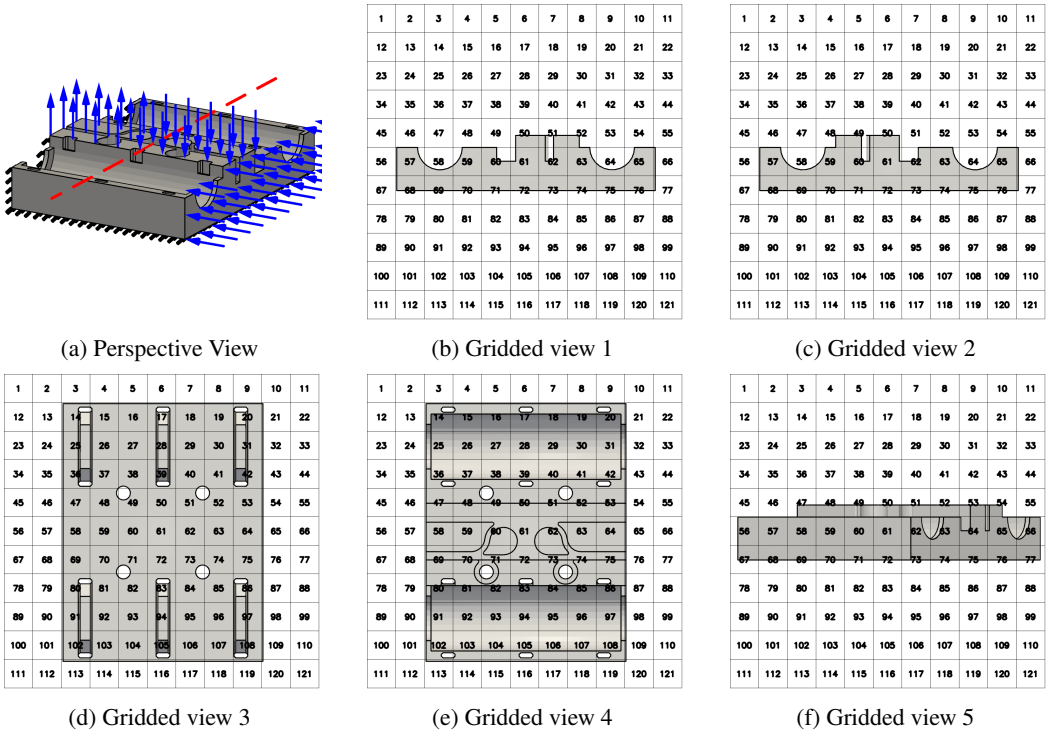
```
***Cells: 24, 25, 26, 27, 28, 29, 30, 31, 32, 35, 36, 37, 38, 39, 40,
41, 42, 43, 46, 47, 48, 49, 50, 51, 52, 53, 54, 57, 58, 59, 60, 61, 62,
```

```
63, 64, 65, 68, 69, 70, 71, 72, 73, 74, 75, 76, 79, 80, 81, 82, 83, 84,
85, 86, 87 ***
***Cells: 46, 47, 48, 49, 50, 51, 52, 53, 54, 57, 58, 59, 60, 61, 62,
63, 64, 65, 68, 69, 70, 71, 72, 73, 74, 75, 76, 79, 80, 81, 82, 83, 84,
85, 86, 87 ***
***Cells: 46, 47, 48, 49, 50, 51, 52, 53, 54, 57, 58, 59, 60, 61, 62,
63, 64, 65, 68, 69, 70, 71, 72, 73, 74, 75, 76, 79, 80, 81, 82, 83, 84,
85, 86, 87 ***
***Cells: 24, 25, 26, 27, 28, 29, 30, 31, 32, 35, 36, 37, 38, 39, 40,
41, 42, 43, 46, 47, 48, 49, 50, 51, 52, 53, 54, 57, 58, 59, 60, 61, 62,
63, 64, 65, 68, 69, 70, 71, 72, 73, 74, 75, 76, 79, 80, 81, 82, 83, 84,
85, 86, 87 ***
***Cells: 47, 48, 49, 50, 51, 52, 53, 58, 59, 60, 61, 62, 63, 64, 69,
70, 71, 72, 73, 74, 75, 80, 81, 82, 83, 84, 85, 86 ***
```

Response: Qwen3 VL 235B A22B Instruct

```
***Cells: 24, 32, 90, 98 ***
***Cells: 35, 43, 79, 87 ***
***Cells: 35, 43, 79, 87 ***
***Cells: 24, 32, 90, 98 ***
***Cells: 35, 43, 79, 87 ***
```

A.4.4 Example Prompt 4: Bending-Compression (*Load+Features+Heuristics*, views=5, Figure 10)



(a) Perspective View (b) Gridded view 1 (c) Gridded view 2

(d) Gridded view 3 (e) Gridded view 4 (f) Gridded view 5

Figure 10: Five images provided jointly with example prompt 4. Top row: non-gridded perspective views for global 3D context. Bottom row: gridded orthographic views used for cell-level region localization.

Response: Gemini 3 Flash Preview

```
 **Image: ('e0', 'a270')**
***I.C.E: 50, 52, 60, 62***
***E.C/P.C: 50, 51, 52, 61, 62***
***T.H: NONE***
***F: 57, 58, 59, 64, 65***

**Image: ('e0', 'a90')**
***I.C.E: 48, 51, 60, 62***
***E.C/P.C: 49, 50, 60, 61***
***T.H: NONE***
***F: 57, 58, 59, 64, 65***

**Image: ('e-90', 'a0')**
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: 14, 25, 36, 17, 28, 39, 20, 31, 42, 49, 52, 71, 74, 80, 91, 102,
83, 94, 105, 86, 97, 108***
***F: NONE***

**Image: ('e90', 'a0')**
***I.C.E: 60, 61, 62, 63***
***E.C/P.C: 60, 61, 62, 63***
***T.H: 4, 6, 8, 41, 43, 49, 52, 71, 74, 78, 80, 82, 113, 116, 119***
***F: 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30, 31, 36, 37, 38,
39, 40, 41, 42, 80, 81, 82, 83, 84, 85, 86, 91, 92, 93, 94, 95, 96, 97,
102, 103, 104, 105, 106, 107, 108***

**Image: ('e0', 'a210')**
***I.C.E: 53, 64***
***E.C/P.C: 47, 48, 49, 50, 51, 52, 53, 54***
***T.H: NONE***
***F: 62, 63, 65, 66***
```

Response: Claude Haiku 4.5

```
# Analysis of Stress-Critical Features

## Image 1: (e0, a270) - Left View
***I.C.E: 57, 58, 59, 64, 65***
***E.C/P.C: 50, 51, 52, 60, 61, 62, 63***
***T.H: 49, 52***
***F: 57, 58, 59, 64, 65***

## Image 2: (e0, a90) - Right View
***I.C.E: 48, 49, 50, 51, 52***
***E.C/P.C: 48, 49, 50, 51, 52***
***T.H: 48, 52***
***F: NONE***

## Image 3: (e-90, a0) - Bottom View
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: 49, 52, 71, 74***
***F: 60, 62, 71, 74***
```

```
## Image 4: (e90, a0) - Top View
***I.C.E: 14, 17, 20, 36, 39, 42, 80, 83, 86***
***E.C/P.C: 14, 17, 20, 36, 39, 42, 80, 83, 86***
***T.H: 49, 52, 71, 74***
***F: 60, 62, 71, 74***

## Image 5: (e0, a210) - Oblique View
***I.C.E: 57, 58, 59, 63, 64, 65***
***E.C/P.C: 50, 51, 52, 60, 61, 62, 63***
***T.H: 49, 52***
***F: 57, 58, 59, 63, 64, 65***
```

Response: GPT 5.4 mini

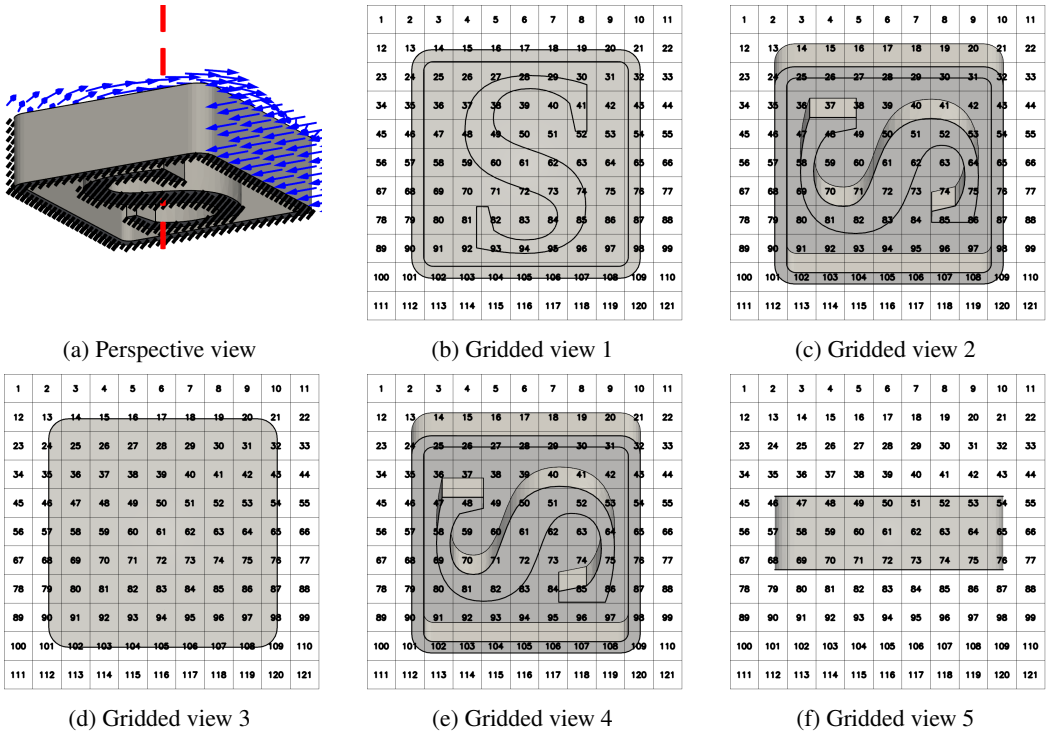
```
***I.C.E: 62, 73, 74, 75 ***
***E.C/P.C: 57, 59, 60, 61, 63, 64, 65, 68, 70, 71, 72, 76, 77 ***
***T.H: 49, 52, 60, 62, 71, 74, 82, 85 ***
***F: 58, 61, 64, 69, 72, 73, 81, 84, 91, 94, 102, 105 ***

***I.C.E: 60, 61, 62, 63, 71, 72, 73, 74, 75, 76, 77 ***
***E.C/P.C: 57, 59, 48, 49, 50, 62, 63, 64, 65, 68, 70, 71, 72, 76, 77,
60, 61, 62, 63, 64, 65, 66 ***
***T.H: 49, 52, 71, 74, 82, 85 ***
***F: 58, 69, 72, 81, 84, 91, 94, 102, 105 ***

***I.C.E: 14, 17, 20, 25, 28, 31, 36, 39, 42, 47, 50, 53, 58, 61, 64,
69, 72, 75, 80, 83, 86, 91, 94, 97, 102, 105, 108 ***
***E.C/P.C: 14, 17, 20, 25, 28, 31, 36, 39, 42, 47, 50, 53, 58, 61, 64,
69, 72, 75, 80, 83, 86, 91, 94, 97, 102, 105, 108 ***
***T.H: 49, 52, 71, 74, 82, 85 ***
***F: NONE ***

***I.C.E: 14, 17, 20, 25, 28, 31, 36, 39, 42, 47, 50, 53, 58, 61, 64,
69, 72, 75, 80, 83, 86, 91, 94, 97, 102, 105, 108 ***
***E.C/P.C: 14, 17, 20, 25, 28, 31, 36, 39, 42, 47, 50, 53, 58, 61, 64,
69, 72, 75, 80, 83, 86, 91, 94, 97, 102, 105, 108 ***
***T.H: 49, 52, 71, 74, 82, 85 ***
***F: 60, 62, 63, 65, 66, 71, 74, 82, 85 ***

***I.C.E: 62, 73, 74, 75 ***
***E.C/P.C: 47, 48, 49, 50, 51, 52, 53, 54, 56, 57, 58, 59, 60, 61, 62,
63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77 ***
***T.H: 63, 66 ***
***F: 62, 64, 65, 73, 74 ***
```

Response: Qwen3 VL 235B A22B Instruct

```
 ***I.C.E: 57, 58, 59, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71, 72, 73,
 74, 75, 76 ***
***E.C/P.C: 50, 51, 52, 61, 62, 63 ***
***T.H: 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77 ***
***F: 57, 58, 59, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71, 72, 73, 74,
75, 76 ***

***I.C.E: 57, 58, 59, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71, 72, 73,
```

```
74, 75, 76 ***
***E.C/P.C: 50, 51, 52, 61, 62, 63 ***
***T.H: 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77 ***
***F: 57, 58, 59, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71, 72, 73, 74,
75, 76 ***

***I.C.E: 14, 17, 20, 80, 83, 86, 102, 105, 108 ***
***E.C/P.C: 14, 17, 20, 80, 83, 86, 102, 105, 108 ***
***T.H: 49, 51, 71, 73 ***
***F: NONE ***

***I.C.E: 14, 17, 20, 80, 83, 86, 102, 105, 108 ***
***E.C/P.C: 14, 17, 20, 80, 83, 86, 102, 105, 108 ***
***T.H: 49, 51, 71, 73 ***
***F: NONE ***

***I.C.E: 57, 58, 59, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71, 72, 73,
74, 75, 76 ***
***E.C/P.C: 50, 51, 52, 61, 62, 63 ***
***T.H: 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77 ***
***F: 57, 58, 59, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71, 72, 73, 74,
75, 76 ***
```

A.4.5 Example Prompt 5: Torsion-Compression (*Load+Features*, views=5, Figure 11)



(a) Perspective view (b) Gridded view 1 (c) Gridded view 2

(d) Gridded view 3 (e) Gridded view 4 (f) Gridded view 5

Figure 11: Five images provided jointly with example prompt 5. Top row: non-gridded perspective views for global 3D context. Bottom row: gridded orthographic views used for cell-level region localization.

Response: Gemini 3 Flash Preview

```
**Image 1 (e-90, a0)**
***I.C.E: NONE***
***E.C/P.C: 27, 28, 29, 30, 31, 38, 39, 40, 41, 42, 49, 50, 51, 52, 53,
60, 61, 62, 63, 64, 71, 72, 73, 74, 75, 82, 83, 84, 85, 86, 93, 94, 95
***
***T.H: NONE***
***F: NONE***

**Image 2 (e-72, a270)**
***I.C.E: 36, 37, 38, 47, 58, 69, 70, 71, 73, 74, 85, 86***
***E.C/P.C: 37, 38, 39, 40, 41, 42, 48, 49, 50, 51, 52, 53, 59, 60, 61,
62, 63, 64, 71, 72, 73, 74, 75, 81, 82, 83, 84, 85***
***T.H: NONE***
***F: 47, 48, 50, 51, 52, 53, 58, 59, 61, 62, 63, 64, 70, 72, 73, 75,
81, 82, 83, 84***

**Image 3 (e90, a0)**
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: NONE***
***F: NONE***

**Image 4 (e-72, a90)**
***I.C.E: 36, 37, 38, 47, 58, 69, 70, 71, 73, 74, 85, 86***
***E.C/P.C: 37, 38, 39, 40, 41, 42, 48, 49, 50, 51, 52, 53, 59, 60, 61,
62, 63, 64, 71, 72, 73, 74, 75, 81, 82, 83, 84, 85***
***T.H: NONE***
***F: 47, 48, 50, 51, 52, 53, 58, 59, 61, 62, 63, 64, 70, 72, 73, 75,
81, 82, 83, 84***

**Image 5 (e0, a0)**
***I.C.E: NONE***
***E.C/P.C: NONE***
***T.H: NONE***
***F: NONE***
```

Response: Claude Haiku 4.5

```
# Analysis of Stress-Critical Features

## Image 1: (e-90, a0) - Bottom View

***I.C.E: 24, 32, 35, 43***
***E.C/P.C: 37, 48, 49, 52, 53***
***T.H: 40, 41, 51, 52, 62, 63***
***F: 40, 41, 51, 52, 62, 63***

## Image 2: (e-72, a270) - Side View

***I.C.E: 24, 25, 32, 33***
***E.C/P.C: 36, 37, 42, 43***
***T.H: NONE***
***F: 25, 26, 31, 32***

## Image 3: (e90, a0) - Top View
```

```
***I.C.E: 14, 21, 24, 32***
***E.C/P.C: NONE***
***T.H: NONE***
***F: 14, 21, 24, 32***

## Image 4: (e-72, a90) - Side View

***I.C.E: 24, 25, 32, 33***
***E.C/P.C: 36, 37, 42, 43, 46, 47, 53, 54***
***T.H: NONE***
***F: 25, 26, 31, 32, 46, 47, 53, 54***

## Image 5: (e0, a0) - Front View

***I.C.E: 46, 54***
***E.C/P.C: NONE***
***T.H: NONE***
***F: 46, 47, 53, 54***
```

Response: GPT 5.4 mini

```
 ***I.C.E: 36, 37, 47, 48, 58, 59, 69, 70, 81, 82, 92, 93 ***
***E.C/P.C: 36, 37, 48, 49, 60, 61, 73, 74, 84, 85, 94, 95 ***
***T.H: NONE ***
***F: 40, 41, 51, 52, 62, 63, 74, 75 ***

***I.C.E: 36, 37, 47, 48, 58, 59, 69, 70, 81, 82, 92, 93 ***
***E.C/P.C: 36, 37, 48, 49, 60, 61, 73, 74, 84, 85, 94, 95 ***
***T.H: NONE ***
***F: 40, 41, 51, 52, 62, 63, 74, 75 ***

***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: NONE ***
***F: NONE ***

***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: NONE ***
***F: NONE ***

***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: NONE ***
***F: NONE ***
```

Response: Qwen3 VL 235B A22B Instruct

```
***I.C.E: 30, 41, 82, 93 ***
***E.C/P.C: 36, 37, 38, 84, 85, 86 ***
***T.H: NONE ***
***F: 14, 21, 102, 109 ***

***I.C.E: 30, 41, 82, 93 ***
***E.C/P.C: 36, 37, 38, 84, 85, 86 ***
***T.H: NONE ***
***F: 14, 21, 102, 109 ***
```

```
***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: NONE ***
***F: 14, 21, 102, 109 ***

***I.C.E: 30, 41, 82, 93 ***
***E.C/P.C: 36, 37, 38, 84, 85, 86 ***
***T.H: NONE ***
***F: 14, 21, 102, 109 ***

***I.C.E: NONE ***
***E.C/P.C: NONE ***
***T.H: NONE ***
***F: NONE ***
```

B CAD Geometry Testset

The evaluation benchmark comprises 156 CAD parts $\times$ 5 load cases (780 paired cases), drawn from the ABC dataset Koch et al. [2019] and FreeCAD Riegel et al. [2016], disjoint from the *orthoViews* training and validation splits. Figures 12–15 show all 156 parts.

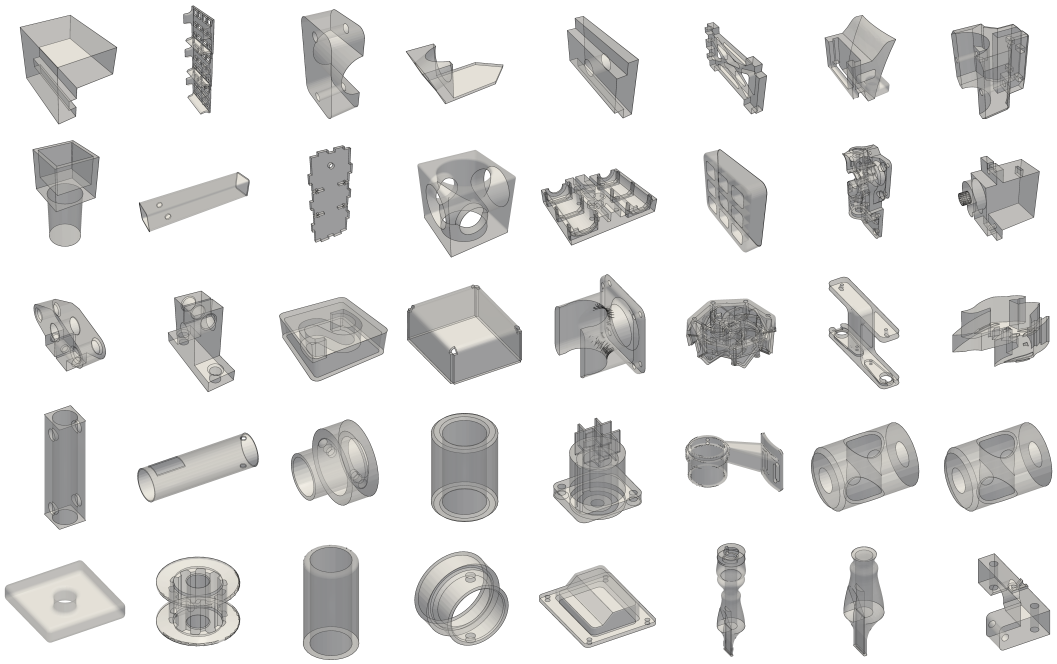


Figure 12: The 156 CAD parts of the evaluation benchmark (part 1 of 4), each rendered under the same semi-transparent orthographic view ($e36, a60$). All parts are disjoint from the *orthoViews* training and validation splits. The first 20 tiles are the 19 benchmark parts from the original submission set followed by the torsion example of Fig. 6; the remaining parts follow in order of object identifier. The set spans extrusions, through-holes, spoked and ribbed bosses, brackets, and filleted load-bearing features.

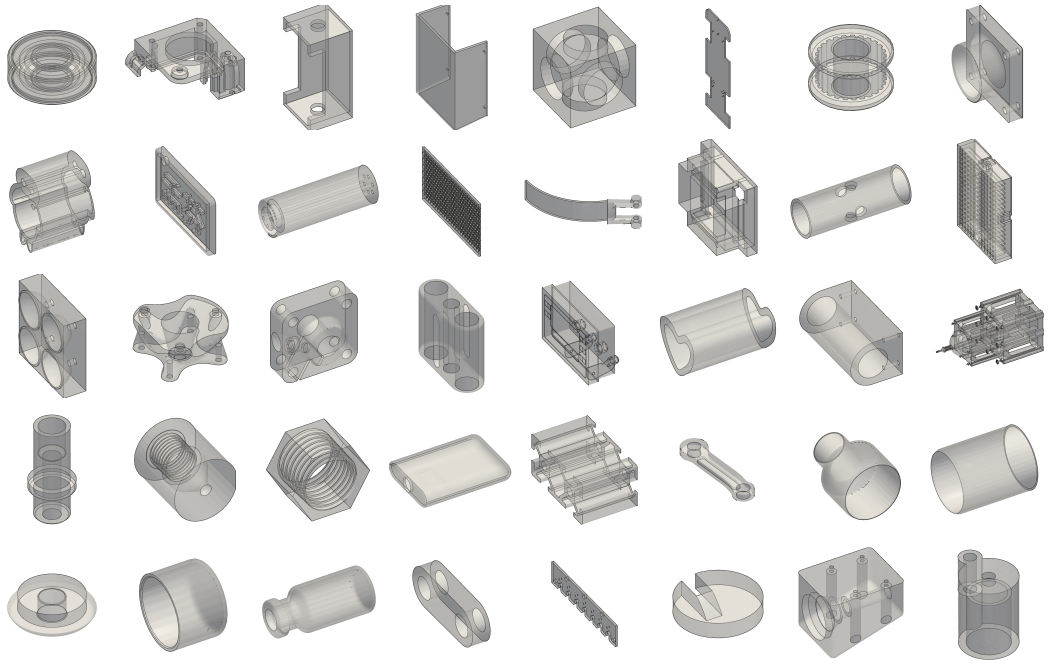


Figure 13: (continued, part 2 of 4) CAD parts of the 156-part evaluation benchmark.

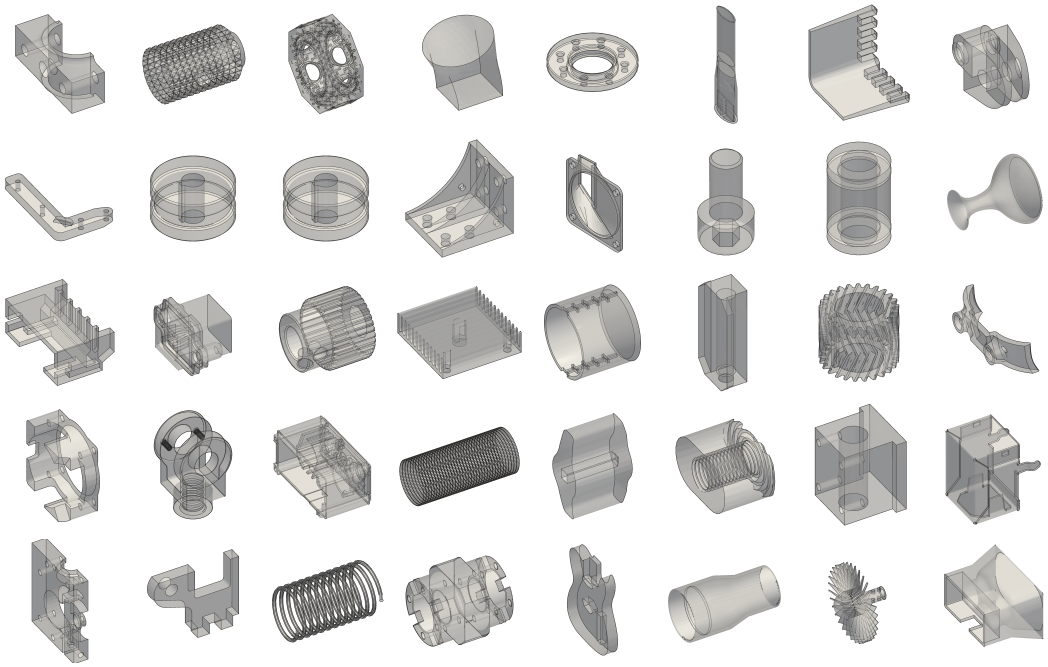


Figure 14: (continued, part 3 of 4) CAD parts of the 156-part evaluation benchmark.

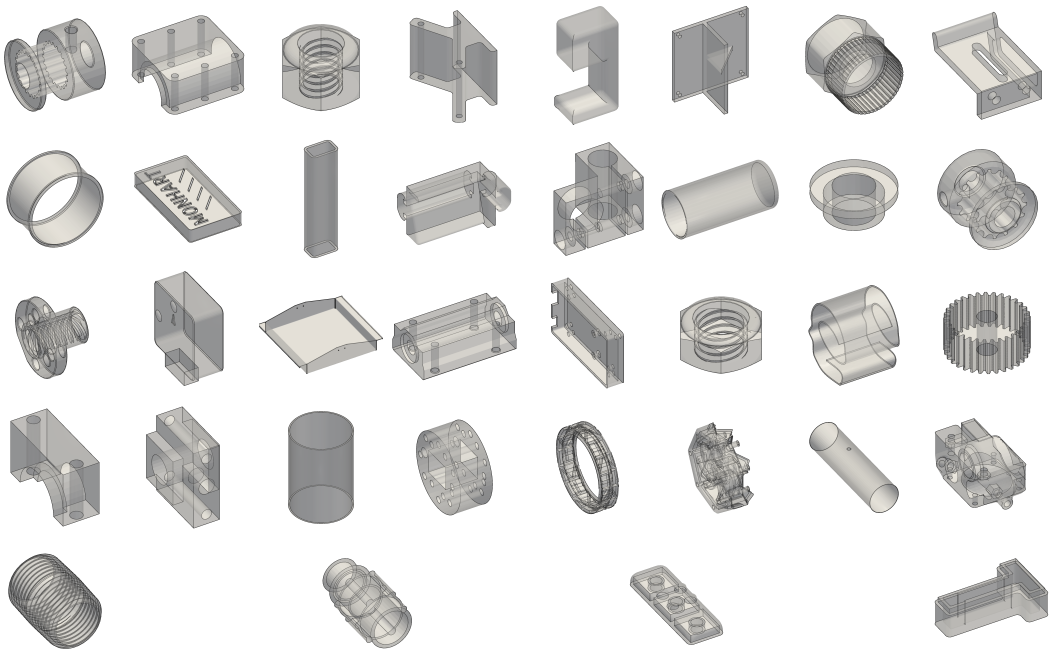


Figure 15: (continued, part 4 of 4) CAD parts of the 156-part evaluation benchmark.

C GReFEM Inference Pipeline

Algorithm 2 GReFEM Zero-Shot Inference Pipeline

Input: 3D Surface Mesh $\mathcal{M}$, Load Case L, Point Density ρ, View count K
Output: 3D Refinement Anchor Points $\mathcal{P}_{3D}$
{**Stage 1: View Selection (*orthoViews*)**}
$\mathcal{V}_{\text{all}} \leftarrow \text{GenerateOrthographicViews}(\mathcal{M})$
$\mathcal{V}_{\text{ranked}} \leftarrow \text{orthoViews}(\mathcal{V}_{\text{all}})$ {Rank via DINOv2 & Self-Attention}
$\mathcal{V}_{\text{top}} \leftarrow \text{SelectTopK}(\mathcal{V}_{\text{ranked}}, K)$
$\mathcal{V}_{\text{grid}} \leftarrow \text{OverlayGrid}(\mathcal{V}_{\text{top}})$
{**Stage 2: Physics-Guided Region Proposal (single MLLM call)**}
$\text{Prompt} \leftarrow \text{ConstructPhysicsPrompt}(L, \text{PromptType})$
$\mathcal{V}_{\text{persp}} \leftarrow \text{RenderPerspectiveViews}(\mathcal{M})$ {Non-gridded views for global 3D context}
$\text{Response} \leftarrow \text{MLLM_Inference}(\text{Prompt}, \mathcal{V}_{\text{persp}} \cup \mathcal{V}_{\text{grid}})$ {All views in one context}
for all $V_k \in \mathcal{V}_{\text{grid}}$ **do**
  $\mathcal{C}_{\text{pred}}[k] \leftarrow \text{ParseSemanticCells}(\text{Response}, k)$ {Feature type $\rightarrow$ cells for view k}
end for
{**Stage 3: 2D Feature Extraction**}
$\mathcal{Q}_{\text{2D}} \leftarrow \emptyset$ {Set of precise 2D pixel coordinates}
for all $V_k \in \mathcal{V}_{\text{grid}}$ **do**
  $\mathcal{E}_{\text{contours}} \leftarrow \text{ExtractGeometricContours}(V_k, \mathcal{M})$
  for all Feature Type $f \in \{\text{ICE, EC/PC, TH, F}\}$ **do**
    $\text{Cells}_f \leftarrow \mathcal{C}_{\text{pred}}[k][f]$
    $\mathcal{Q}_f \leftarrow \text{IntersectContoursWithCells}(\mathcal{E}_{\text{contours}}, \text{Cells}_f)$
    $\mathcal{Q}_{\text{2D}} \leftarrow \mathcal{Q}_{\text{2D}} \cup \mathcal{Q}_f$
  end for
end for
{**Stage 4: 3D Spatial Grounding (MV-RaySeg)**}
$\mathcal{P}_{\text{raw}} \leftarrow \emptyset$
for all $V_k \in \mathcal{V}_{\text{grid}}$ and corresponding $\mathcal{Q}_{\text{2D}}$ **do**
  $\mathbf{d}_v, \mathbf{c}_v \leftarrow \text{GetCameraParams}(V_k)$
  for all $\mathbf{q} \in \mathcal{Q}_{\text{2D}}$ **do**
    $T_q \leftarrow \text{RayMeshIntersection}(\mathbf{q}, \mathbf{d}_v, \mathbf{c}_v, \mathcal{M})$
    if Feature is Surface-Visible (e.g., EC/PC) **then**
      $\mathcal{P}_{\text{raw}} \leftarrow \mathcal{P}_{\text{raw}} \cup \{T_{q,0}\}$ {Keep first hit only}
    else
      {Deep volumetric feature (ICE, F, TH)}
      for $i = 0$ to $|T_q| - 1$ step 2 **do**
        $\mathcal{P}_{\text{seg}} \leftarrow \text{SampleSegment}(T_{q,i}, T_{q,i+1}, \rho)$
        $\mathcal{P}_{\text{raw}} \leftarrow \mathcal{P}_{\text{raw}} \cup \mathcal{P}_{\text{seg}}$
      end for
    end if
  end for
end for
$\mathcal{P}_{3D} \leftarrow \text{DensityFilter}(\mathcal{P}_{\text{raw}}, \rho)$ {Remove redundant overlapping points}
return $\mathcal{P}_{3D}$

D Mesh Geometric Heuristic Pipeline

Algorithm 3 Mesh Geometric Heuristic Baseline

Input: 3D Surface Mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, Angle Threshold θ_{thresh} (e.g., 30°), Spatial Density ρ
Output: Dense Geometric Refinement Points $\mathcal{P}_{\text{base}}$
{**Stage 1: Process Mesh Connectivity**}
Extract adjacent face pairs (f_1, f_2) sharing an edge e
Compute face normals $\mathbf{n}_1, \mathbf{n}_2$ and centroids $\mathbf{c}_1, \mathbf{c}_2$ for all faces in $\mathcal{F}$
$\mathcal{P}_{\text{sharp}} \leftarrow \emptyset, \quad \mathcal{F}_{\text{smooth}} \leftarrow \emptyset$
{**Stage 2: Extract and Upsample Sharp Concave Edges**}
for all adjacent pairs (f_1, f_2) **do**
  $\theta \leftarrow \arccos(\mathbf{n}_1 \cdot \mathbf{n}_2)$
  if $\theta > \theta_{\text{thresh}}$ **then**
    $\mathbf{v}_{\text{dir}} \leftarrow \mathbf{c}_2 - \mathbf{c}_1$
    if $\mathbf{v}_{\text{dir}} \cdot \mathbf{n}_1 > 0$ **then**
      Retrieve vertices $\mathbf{v}_a, \mathbf{v}_b$ of edge e {Edge is concave (inward bending)}
      $L_e \leftarrow \|\mathbf{v}_b - \mathbf{v}_a\|_2$
      $N_{\text{samples}} \leftarrow \max(2, \lceil L_e/\rho \rceil)$
      Sample N_{samples} equidistant points along segment $[\mathbf{v}_a, \mathbf{v}_b]$
      $\mathcal{P}_{\text{sharp}} \leftarrow \mathcal{P}_{\text{sharp}} \cup \{\text{sampled points}\}$
    end if
  end if
end for
{**Stage 3: Extract and Sample Smooth Concave Surfaces (Fillets/Holes)**}
for all adjacent pairs (f_1, f_2) **do**
  $\theta \leftarrow \arccos(\mathbf{n}_1 \cdot \mathbf{n}_2)$
  if $1^\circ < \theta \leq \theta_{\text{thresh}}$ **then**
    $\mathbf{v}_{\text{dir}} \leftarrow \mathbf{c}_2 - \mathbf{c}_1$
    if $\mathbf{v}_{\text{dir}} \cdot \mathbf{n}_1 > 0$ **then**
      $\mathcal{F}_{\text{smooth}} \leftarrow \mathcal{F}_{\text{smooth}} \cup \{f_1, f_2\}$ {Smooth concavity}
    end if
  end if
end for
$A_{\text{total}} \leftarrow \sum_{f \in \mathcal{F}_{\text{smooth}}} \text{Area}(f)$
$N_{\text{total_samples}} \leftarrow \lceil A_{\text{total}}/\rho^2 \rceil$
$\mathcal{P}_{\text{smooth}} \leftarrow \emptyset$
for $i = 1$ to $N_{\text{total_samples}}$ **do**
  Select face $f \in \mathcal{F}_{\text{smooth}}$ with probability $\propto \text{Area}(f)/A_{\text{total}}$
  Generate random barycentric coordinates (u, v, w) {Uniform triangle sampling}
  Compute 3D point $\mathbf{p} = u\mathbf{v}_0 + v\mathbf{v}_1 + w\mathbf{v}_2$ using vertices of f
  $\mathcal{P}_{\text{smooth}} \leftarrow \mathcal{P}_{\text{smooth}} \cup \{\mathbf{p}\}$
end for
{**Stage 4: Combine and Filter**}
$\mathcal{P}_{\text{base}} \leftarrow \mathcal{P}_{\text{sharp}} \cup \mathcal{P}_{\text{smooth}}$
$\mathcal{P}_{\text{base}} \leftarrow \text{DensityFilter}(\mathcal{P}_{\text{base}}, \rho)$ {Remove redundant overlap}
return $\mathcal{P}_{\text{base}}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to Section 5.1 and Table 1, where the precision and recall frontiers mentioned in the introduction are discussed for **GReFEM**, demonstrating that MLLMs act as precise zero-shot geometric assistants across various loading cases.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Section 6 (Conclusion and Limitations), where we explicitly discuss the current limitations of MLLMs, particularly their lag in raw recall compared to blind geometric heuristics and the open challenge of end-to-end autonomous 3D physical reasoning.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This is an exploratory empirical study evaluating the zero-shot capabilities of existing foundation models. It does not introduce new theoretical bounds or mathematical proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully disclose the exact prompt templates used for all MLLMs (Appendix A), the mesh generation distance-field parameters in Section 5, CAD geometries (Appendix B), GreFem inference pipeline pseudocode (Appendix C) and Mesh Geoemtric Heurisitic psuedocode (Appendix D).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We now open-source the full codebase (Inference, CV tools, FEniCSx and Gmsh pipelines, CAD dataset to train the view selector *orthoViews*) on our github repo https://github.com/kbali1297/GReFEM.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details, including the matched-budget element sizing parameters, loading conditions, and LLM inference setups, are specified in Section 5 (Experimental Setup) and Appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard deviations of Downstream relative FEM errors, precision, recall, and F1 scores across multiple experimental dimensions (MLLMs, load cases, prompts, and number of views) in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We document the use of Openrouter for access to MLLMs listed in the paper and the Nvidia H100 GPU used to train the view selector on the CAD geometry dataset.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research, which focuses on computational geometry and finite element analysis, fully conforms to these guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Our work focuses on foundational computational engineering and optimizing finite element discretization. We do not foresee any direct negative societal applications or malicious misuse arising from automating CAD mesh generation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This paper explores the use of existing commercial MLLM APIs for mechanical engineering tasks and does not release new high-risk foundation models or scraped internet datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite the creators of the ABC CAD dataset and FreeCAD geometries in Section 5. The open-source availability and usage terms are fully respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce a curated evaluation benchmark of CAD objects paired with multi-axial physics-based prompt templates, which is documented and provided in the anonymized supplementary material. We also visually provide all the CAD samples on which the studies have been done in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our research evaluates foundation models on synthetic CAD geometries and does not involve human subjects or crowdsourcing.

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This research does not involve human subjects, making IRB approval not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The core methodology of this research explicitly evaluates the zero-shot spatial reasoning capabilities of MLLMs (Gemini, Claude, GPT, Qwen) for mesh refinement. Their exact usage, visual inputs, and inference setup are thoroughly described in Sections 4 and 5, with prompt templates provided in the Appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.