Article | Open Access

Journal of Mechanoinformatics

Lin *et al. J. Mechanoinform.* **2026**, *1*, 2 **DOI:**10.20517/jmechinfo.2026.03

TO-Master: an LLM-agent framework for automated topology optimization

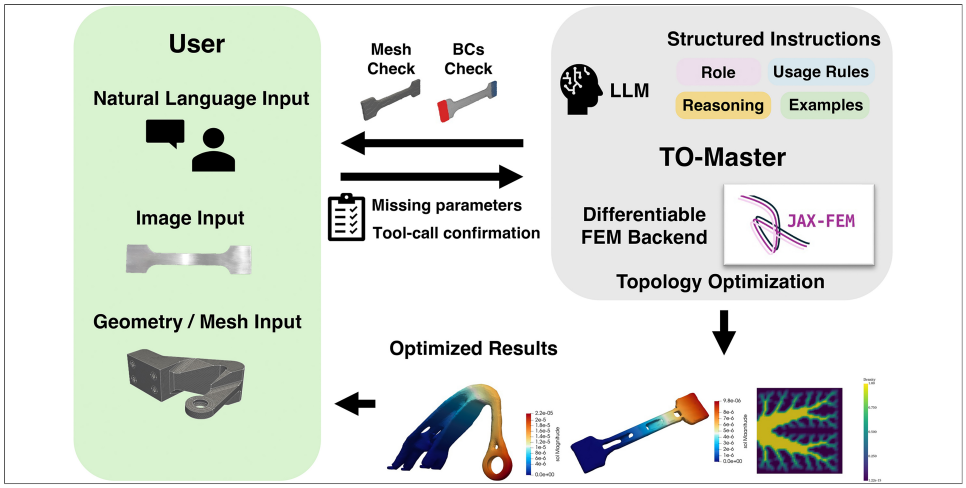
Haoju Lin[1], Wenchang Zhang[1], Weipeng Xu[1], Xiang Li[2], Tian Xu[1], Tianju Xue[1]

Keywords:
Topology optimization, LLM agent, natural-language-driven modeling, automated model generation

Citation: Lin, H.; Zhang, W.; Xu, W.; Li, X.; Xu, T.; Xue, T. TO-Master: an LLM-agent framework for automated topology optimization. *J. Mechanoinform.* **2026**, *1*, 2. https://dx.doi.org/10.20517/jmechinfo.2026.03

Received: 25 Aug 2026 **First Decision:** 16 Oct 2026 **Revised:** 5 Nov 2026 **Accepted:** 13 Nov 2026 **Published:** 27 Nov 2026

Academic Editor: Tienchong Chang **Copy Editor:** Ting-Ting Hu **Production Editor:** Ting-Ting Hu



Abstract

Topology optimization (TO) has become a mature computational design method. However, many TO workflows still require manual effort in geometry and mesh preparation, boundary-condition assignment, solver setup, and post-processing. These requirements can remain a barrier for users who do not already work within a specialized computer-aided design (CAD) or simulation environment. This work introduces TO-Master, a large language model (LLM) agent framework that turns finite-element-based TO into a conversational, tool-orchestrated workflow. From natural language instructions and optional mesh, geometry, or image inputs, the agent selects computational tools, constructs finite element TO models, checks meshes and boundary conditions, and launches sensitivity-based optimization with typed solver arguments. The framework supports generated and uploaded meshes, image-to-mesh conversion, two-dimensional (2D) and three-dimensional (3D) structural compliance minimization, thermal conduction, multiple load cases, stress-constrained optimization, and engineering geometries. Numerical experiments show that TO-Master can reproduce standard benchmark results and solve more complex engineering examples while returning optimized results, field distributions, convergence histories, and interactive artifacts without user-written code. An instruction ablation study evaluates the effects of tool-interface documentation, internal reasoning guidance, and few-shot examples on formulation success under ambiguous user

Check for updates

[1]Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.
[2]China Iron and Steel Research Institute Group, Beijing 100081, China.

Correspondence to: Dr. Tian Xu, Prof. Tianju Xue, Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. E-mail: tianxu@ust.hk; cetxue@ust.hk

specifications. TO-Master enables automated TO problem formulation and execution across complex geometries and diverse problem types by combining LLM-based problem formulation, explicit mesh and boundary-condition verification, and predefined physics-based computational tools. The TO-Master platform is available online at https://www.bohrium.com/en/apps/to-master.

INTRODUCTION

Topology optimization (TO) is a computational design method that optimizes the material distribution in a prescribed design domain subject to boundary conditions, constraints, and objective functions. Classical methods include the homogenization-based method[1], the solid isotropic material with penalization (SIMP) method[2-4], the level-set method[5,6], and the bi-directional evolutionary structural optimization method[7,8]. These methods have been widely used for structural design, thermal design, and multi-physics inverse design problems. However, setting up a TO problem remains a nontrivial task. Users must specify geometries, meshes, boundary conditions, material models, objectives, constraints, optimization parameters, and postprocessing procedures. This process usually requires substantial problem-specific scripting and repeated manual modification, which limits the accessibility of TO workflows.

Recent advances in differentiable programming and automatic differentiation provide new opportunities for computational mechanics and inverse design. Automatic-differentiation-based TO frameworks such as AuTO[9] or differentiable finite element method (FEM) frameworks such as JAX-FEM[10] show that finite element assembly, solution procedures, and objective evaluations can be embedded in automatic-differentiation-compatible computational graphs, making it possible to obtain sensitivities through automatic differentiation. These developments reduce the implementation burden of sensitivity analysis. Nevertheless, differentiable solvers still require users to formulate the computational problem correctly. In practice, the main difficulty often lies not only in solving the optimization problem constrained by partial differential equations (PDEs), but also in translating an engineering design intent into a complete and valid solver input.

Large language models (LLMs) have shown strong capabilities in language understanding, code generation, tool use, and multi-step reasoning. Chain-of-thought prompting improves reasoning over intermediate steps[11], while agentic frameworks such as ReAct enable LLMs to interact with external tools during problem solving[12]. Tool-augmented language models further demonstrate that LLMs can call external application programming interfaces (APIs) and use computational tools as part of their reasoning process[13]. These capabilities make LLM agents promising interfaces for scientific computing workflows, where user intent must be mapped to a sequence of structured computational operations. In computer-aided engineering (CAE), LLM agents have been explored for LLM-empowered CAE frameworks[14], automated structural analysis[15], computational fluid dynamics (CFD) case construction and solver execution[16,17], and text-driven computer-aided design (CAD) generation[18,19]. These studies show that LLM agents can reduce manual intervention in simulation setup and workflow orchestration. Related human-in-the-loop TO studies have incorporated user preferences through interactive feature-size control, user-drawn patterns, and AI-guided region selection[20-23]. Recent and concurrent studies have also connected LLM agents with topology optimization, including LLM-based controllers for adapting SIMP continuation parameters[24], AutoSiMP for mapping natural-language descriptions to SIMP topology optimization problems[25], TO-Agents for preference-guided topology optimization with multi-agent visual feedback[26], and large-model-assisted conceptual structural design realization[27]. Park and Chung proposed TopOptAgents, a multi-agent framework that combines problem formulation, code generation, and result assessment through iterative self-refinement[28].

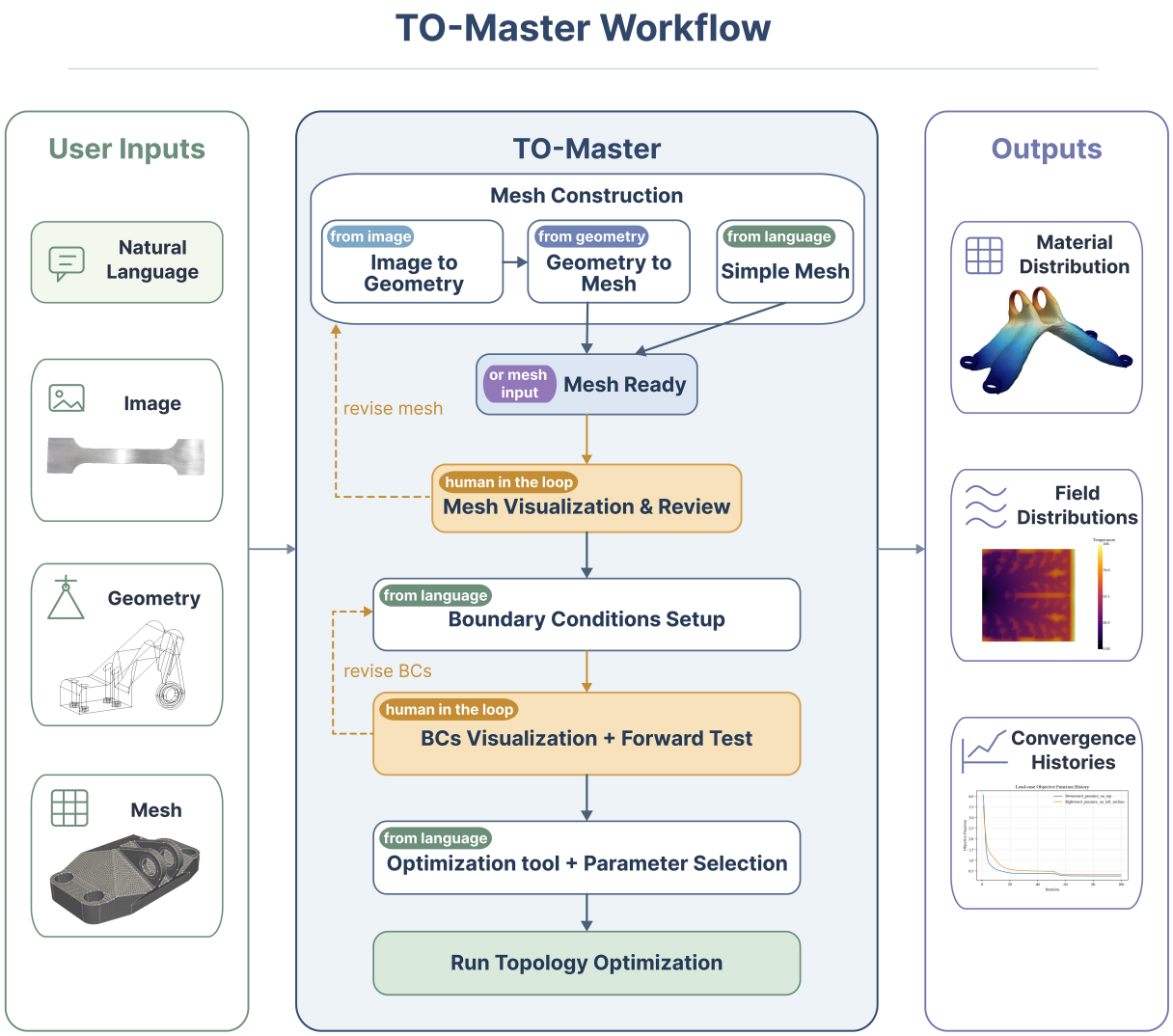


Figure 1. End-to-end workflow of the TO-Master framework. User-level natural language specifications are optionally supplemented by mesh files, geometry files, or image inputs, and are translated into FEM-based topology optimization problems. The framework builds or imports mesh models, visualizes and verifies meshes and boundary conditions through human-in-the-loop confirmation, performs sensitivity-based optimization, and returns optimized results, field distributions, and convergence histories. BCs: Boundary conditions; FEM: finite element method; TO: topology optimization.

However, integrating diverse TO formulations with complex three-dimensional (3D) geometries remains underexplored in LLM-agent frameworks for TO, particularly when unstructured meshes and image-based model generation are required.

In this work, we address this challenge with TO-Master, an LLM-agent framework that combines automated problem formulation, mesh and boundary-condition verification, and predefined numerical tools. The framework supports image-to-mesh conversion and 3D unstructured engineering meshes, together with structural compliance minimization, thermal conduction, multiple load cases, and stress-constrained optimization. As shown in Figure 1, it translates natural-language specifications, optionally supplemented by mesh, geometry, or image files, into finite element TO models. The workflow builds or imports meshes, visualizes and checks meshes and boundary conditions through human-in-the-loop confirmation, performs sensitivity-based optimization, and returns optimized results, field distributions, and convergence histories. The differentiable JAX-FEM backend performs the finite element analyses and sensitivity calculations, while the agent coordinates problem formulation and tool execution. For the same problem and settings, numerical optimization retains the computational efficiency of the underlying implementation. We evaluate the framework using benchmark and extended cases, together with an instruction ablation study.

Table 1. Comparison of LLM-agent frameworks for TO

Aspect	LLM controller[24]	AutoSiMP[25]	TO-Agents[26]	TopOptAgents[28]	TO-Master
LLM role	Continuation control	Setup; control	Preference-guided refinement	Formulation; code generation	Formulation; orchestration
Solver separation	Predefined solver	Predefined solver	Predefined solver	LLM-generated code	Predefined solver
Differentiable backend	-	-	-	✓	✓
3D unstructured meshes	-	-	Backend support	-	✓
Image-to-mesh	-	-	-	-	✓
Conventional compliance minimization	✓	✓	✓	✓	✓
Other TO problems	-	-	-	Stress minimization	Thermal; multi-load; stress-constrained
LLM adaptive control	✓	✓	✓	✓	-
Verification	Validity; grayness	Inputs; structure	Inputs; visual feedback	Inputs; code; results	Mesh; boundary conditions

3D: Three-dimensional; LLM: large language model; TO: topology optimization.

To further emphasize the advantages of the proposed TO-Master framework, Table 1 compares its computational workflow and demonstrated capabilities with those of related LLM-agent frameworks for TO. The LLM continuation controller[24] and AutoSiMP[25] offer LLM-based parameter adaptation during optimization, whereas TO-Master performs the numerical iterations without LLM intervention. In its interactive workflow, TO-Master requires user confirmation of the proposed settings before launching an optimization or rerunning it with modified parameters. This design is intended to limit unnecessary, computationally expensive reruns and keep computational expenditure under user control.

The remainder of this paper is organized as follows. Section "EXPERIMENTAL" describes the implemented density-based topology optimization formulations and the TO-Master framework, including the differentiable FEM backend, tool interface, prompt design, and agent workflow. Section "RESULTS AND DISCUSSION" presents the benchmark and engineering examples, reports the prompt ablation study, and discusses the effectiveness, robustness, and limitations of the proposed framework. Section "CONCLUSIONS" summarizes the main conclusions and future research directions.

EXPERIMENTAL

Problem formulation for topology optimization

This section summarizes the TO formulations implemented in TO-Master. We first introduce the baseline density-based structural compliance minimization problem, and then describe the density filtering, projection, passive-domain treatment, and problem-specific extensions used in the implementation.

Baseline structural compliance minimization

The design domain Ω is discretized into finite elements Ω_e, $e = 1, \ldots, N_e$. In density-based TO, each element is assigned a scalar density variable $\rho_e \in [0,1]$, where $\rho_e = 0$ represents void and $\rho_e = 1$ represents solid material. The design variables are collected as $\rho = \{\rho_e\}_{e=1}^{N_e}$.

For structural optimization, the displacement field is approximated as

$$\mathbf{u}^h(\mathbf{x}) = \sum_{k=1}^{N_n} \phi_k(\mathbf{x})\mathbf{U}_k, \tag{1}$$

where φ_k is the finite element shape function associated with node k, U_k is the nodal displacement vector, and N_n is the number of nodes. The weak form is to find $u^h \in V_g$ such that

$$\int_\Omega \boldsymbol{\sigma}(\mathbf{u}^h, \boldsymbol{\rho}) : \boldsymbol{\varepsilon}(\mathbf{v})\, d\Omega = \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{v}\, d\Gamma, \qquad \forall \mathbf{v} \in V_0, \tag{2}$$

where V_g satisfies the prescribed displacement boundary conditions, V_0 is the corresponding test space with homogeneous essential boundary conditions, t is the applied traction, and ε is the small-strain tensor.

The SIMP interpolation is used to relate density to stiffness[4,29],

$$E_e = E_{\min} + (E_{\max} - E_{\min})\rho_e^p, \tag{3}$$

where $E_{\max}$ and $E_{\min}$ are the solid and ersatz Young's moduli, and p is the penalization factor.

The material usage is controlled by the standard volume-fraction constraint,

$$\frac{\sum_{e=1}^{N_e} \rho_e |\Omega_e|}{\sum_{e=1}^{N_e} |\Omega_e|} \le V_f, \tag{4}$$

where $|\Omega_e|$ denotes the element area in a two-dimensional (2D) domain or volume in 3D, and V_f is the prescribed material volume fraction. In the implemented structural optimization tools, this constraint is discretized as the mean density over the optimized element set.

The baseline structural compliance minimization problem is written as

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & J(\boldsymbol{\rho}) = \mathbf{F}^T\mathbf{U}, \\ \text{s.t.} \quad & \mathbf{K}(\boldsymbol{\rho})\mathbf{U} = \mathbf{F}, \\ & \frac{\sum_{e=1}^{N_e} \rho_e |\Omega_e|}{\sum_{e=1}^{N_e} |\Omega_e|} \le V_f, \\ 0 \le \rho_e \le 1, \quad & e = 1, \ldots, N_e. \end{aligned} \tag{5}$$

Density filtering, projection, and passive domains

In the implementation, the raw density field is modified by density filtering, optional Heaviside projection, and passive-domain treatment before finite element analysis. Density filtering is used to regularize the design field and reduce mesh-dependent numerical instabilities[30]. When a problem contains passive domains, only the elements outside the passive domains are treated as designable. The set of designable elements is denoted by $\mathcal{I}_d$, and the passive elements are denoted by $\mathcal{I}_p$. The raw design variables are then $\rho = \{\rho_e\}_{e \in \mathcal{I}_d}$.

The density filter is applied to the designable elements:

$$\bar{\rho}_e = \frac{\sum_{j \in \mathcal{N}_e} w_{ej}\rho_j}{\sum_{j \in \mathcal{N}_e} w_{ej}}, \qquad w_{ej} = \max(0, r_{\min} - d_{ej}), \tag{6}$$

where d_{ej} is the distance between element centroids and $\mathcal{N}_e$ is the set of neighboring designable elements within the filter radius. The filter radius is defined as

$$r_{\min} = r_f \bar{h}, \tag{7}$$

where r_f is the prescribed filter factor and $\bar{h}$ is the average element size. In the implementation, $\bar{h}$ is computed from the average cell measure over the full finite element mesh,

$$\bar{h} = \left(\frac{1}{N_e} \sum_{e=1}^{N_e} |\Omega_e| \right)^{1/d}, \tag{8}$$

where d is the spatial dimension. Thus, r_f is a dimensionless factor controlling the filter radius relative to the mesh size.

When projection is enabled, the filtered density is further mapped to the physical density field $\widetilde{\rho}$ through Heaviside-type projection[31,32],

$$\widetilde{\rho}_e = \frac{\tanh(\beta\eta) + \tanh[\beta(\bar{\rho}_e - \eta)]}{\tanh(\beta\eta) + \tanh[\beta(1-\eta)]}, \tag{9}$$

where β controls the projection sharpness and η is the projection threshold. If projection is not used, $\widetilde{\rho}_e = \bar{\rho}_e$.

The full element density field used in finite element analysis (FEA) is

$$\rho_e^{\text{full}} = \begin{cases} \widetilde{\rho}_e, & e \in \mathcal{I}_d, \\ 1, & e \in \mathcal{I}_p. \end{cases} \tag{10}$$

Therefore, ρ is updated by the optimizer, while $\widetilde{\rho}$ defines the physical densities on the designable elements. The assembled field ρ^{full} defines the material distribution over the whole mesh and is used in FEA and objective evaluation. For structural problems with passive domains, the structural tools apply the implemented material constraint over the designable elements:

$$\frac{1}{|\mathcal{I}_d|} \sum_{e \in \mathcal{I}_d} \widetilde{\rho}_e \le V_f. \tag{11}$$

Multi-load-case topology optimization

For problems with multiple load cases, the same density field is shared by all load cases. The objective is the sum of compliances,

$$J_{\text{multi}}(\boldsymbol{\rho}) = \sum_{m=1}^{N_L} \mathbf{F}_m^T \mathbf{U}_m, \qquad \mathbf{K}(\boldsymbol{\rho}^{\text{full}})\mathbf{U}_m = \mathbf{F}_m, \tag{12}$$

where N_L is the number of load cases. The material-volume constraint is imposed on the densities of the designable elements using Equation (11).

Stress-constrained topology optimization

For stress-constrained problems, TO-Master uses a differentiable aggregated stress constraint. Directly constraining the maximum von Mises stress is numerically inconvenient because the maximum operator is non-smooth and stresses in low-density elements may introduce singular or noisy sensitivities. To obtain a stable constraint for gradient-based optimization, the stress is evaluated using a density-weighted stress measure and then aggregated by a p-norm, following the maximum-stress-measure strategy used in stress-constrained TO[33].

The density-weighted stress is computed with

$$E_e^{\sigma} = E_{\max} \left(\rho_e^{\text{full}} \right)^{p_\sigma}, \tag{13}$$

$$\widetilde{\rho}$$

where $p_\sigma = 0.5$ in the implementation. This weighting reduces the influence of low-density regions in the stress constraint while retaining a smooth dependence on the design variables. The elemental stress value is defined as the element-averaged density-weighted von Mises stress,

$$\bar{\sigma}_{vm,e} = \frac{\int_{\Omega_e} \sigma_{vm}(\mathbf{x})\, d\Omega}{\int_{\Omega_e} d\Omega}. \tag{14}$$

The element average reduces quadrature-point noise and provides one representative stress value for each element.

The global stress measure is approximated by

$$P_\sigma = \left[\sum_e \left(\frac{\bar{\sigma}_{vm,e}}{\sigma_{allow}} \right)^{p_s} \right]^{1/p_s}, \tag{15}$$

where $p_s = 12$ and σ_{allow} is the prescribed allowable stress. To improve the approximation to the maximum stress, an iteration-dependent scaling factor is applied based on the adaptive normalization approach of Le *et al.*[34]:

$$c_\sigma^{(k)} = q_n \frac{\sigma_{\max}^{(k-1)}}{\sigma_{allow} P_\sigma^{(k-1)}} + (1 - q_n) c_\sigma^{(k-1)}, \qquad q_n = 0.5. \tag{16}$$

The stress constraint is then written as

$$g_\sigma(\boldsymbol{\rho}) = c_\sigma P_\sigma - 1 \le 0. \tag{17}$$

The complete stress-constrained problem is

$$\begin{aligned} \min_{\rho} \quad & J(\boldsymbol{\rho}) = \mathbf{F}^T \mathbf{U}, \\ \text{s.t.} \quad & \mathbf{K}(\boldsymbol{\rho}^{\text{full}})\mathbf{U} = \mathbf{F}, \\ & \frac{1}{|\mathcal{I}_d|} \sum_{e \in \mathcal{I}_d} \widetilde{\rho}_e \le V_f, \\ & g_\sigma(\boldsymbol{\rho}) \le 0, \\ & 0 \le \rho_e \le 1, \qquad e \in \mathcal{I}_d. \end{aligned} \tag{18}$$

After optimization, TO-Master also evaluates the von Mises stress on the thresholded solid topology for validation and visualization. This post-processed solid-material stress is not used as the optimization constraint.

Thermal topology optimization

For thermal conduction problems, TO-Master solves the steady-state heat conduction equation in weak form: find $T^h \in V_T$ such that

$$\int_\Omega k(\rho^{\text{full}}) \nabla T^h \cdot \nabla v \, d\Omega = \int_\Omega q v \, d\Omega + \int_{\Gamma_q} \bar{q} v \, d\Gamma, \qquad \forall v \in V_0, \tag{19}$$

where T^h is the finite element temperature field, q is the volumetric heat source, V_T satisfies the prescribed temperature boundary conditions on Γ_T, V_0 is the corresponding test space with homogeneous temperature conditions on Γ_T, and $\bar{q}$ is the prescribed heat flux on Γ_q. Boundary segments without a prescribed temperature or heat flux are treated as insulated boundaries, i.e., $\bar{q} = 0$. The effective thermal conductivity is interpolated as

$$k_e = k_{\min} + (k_{\max} - k_{\min}) \left(\rho_e^{\text{full}} \right)^p, \tag{20}$$

where $k_{\max}$ and $k_{\min}$ are the solid and ersatz conductivities.

The thermal objective is the heat-source-weighted temperature integral,

$$J_T(\boldsymbol{\rho}) = \int_\Omega qT \, d\Omega. \tag{21}$$

This objective follows the standard thermal compliance-type formulation used in heat-conduction TO[35].

The thermal optimization problem is written as

$$\begin{aligned} &\min_{\rho} \quad J_T(\rho), \\ \text{s.t.} \quad &-\nabla \cdot \left[k(\rho^{\text{full}})\nabla T\right] = q, \\ &\frac{1}{|\mathcal{I}_d|}\sum_{e\in\mathcal{I}_d} \widetilde{\rho}_e \le V_f, \\ &0 \le \rho_e \le 1, \qquad e \in \mathcal{I}_d. \end{aligned} \tag{22}$$

Sensitivity analysis and gradient-based optimization

The TO problems described above are solved as PDE-constrained optimization problems. Following the discretize-then-optimize strategy used in JAX-FEM[10], the governing equations are first discretized by FEM, and the resulting finite-dimensional optimization problem is then solved. To cover both structural and thermal problems, the discrete state variable is denoted by s, where $s = U$ for structural analysis and $s = T$ for thermal conduction. The discretized governing equation is written as

$$\mathbf{C}(\mathbf{s}, \boldsymbol{\rho}) = \mathbf{0}, \tag{23}$$

where C is the assembled residual after imposing the prescribed boundary conditions.

The solution of Equation (23) defines the state as an implicit function of the design, $s = s(\rho)$. The reduced objective and constraints are

$$\widehat{J}(\boldsymbol{\rho}) = J\left(\mathbf{s}(\boldsymbol{\rho}), \boldsymbol{\rho}\right), \qquad \widehat{g}_i(\boldsymbol{\rho}) = g_i\left(\mathbf{s}(\boldsymbol{\rho}), \boldsymbol{\rho}\right), \tag{24}$$

and the reduced TO problem is

$$\begin{aligned} &\min_{\rho\in[0,1]^{N_\rho}} \quad \widehat{J}(\boldsymbol{\rho}), \\ \text{s.t.} \quad &\widehat{g}_i(\boldsymbol{\rho}) \le 0, \qquad i = 1, \ldots, N_g, \end{aligned} \tag{25}$$

where N_ρ is the number of design variables and N_g is the number of inequality constraints.

For gradient-based optimization, the derivative of the reduced objective is computed by differentiating the residual equation. The chain rule gives

$$\frac{d\widehat{J}}{d\boldsymbol{\rho}} = \frac{\partial J}{\partial \mathbf{s}}\frac{d\mathbf{s}}{d\boldsymbol{\rho}} + \frac{\partial J}{\partial \boldsymbol{\rho}}, \tag{26}$$

while differentiating $C\left(s\left(\rho\right), \rho\right) = 0$ yields

$$\frac{\partial \mathbf{C}}{\partial \mathbf{s}}\frac{d\mathbf{s}}{d\boldsymbol{\rho}} + \frac{\partial \mathbf{C}}{\partial \boldsymbol{\rho}} = \mathbf{0}. \tag{27}$$

Since the number of design variables is typically large in TO, the adjoint form is used for efficiency. Introducing the adjoint variable λ, we solve

$$\left(\frac{\partial \mathbf{C}}{\partial \mathbf{s}}\right)^T \boldsymbol{\lambda} = \left(\frac{\partial J}{\partial \mathbf{s}}\right)^T, \tag{28}$$

and obtain

$$\frac{d\widehat{J}}{d\boldsymbol{\rho}} = \frac{\partial J}{\partial \boldsymbol{\rho}} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{C}}{\partial \boldsymbol{\rho}}. \tag{29}$$

The same adjoint procedure is applied to state-dependent constraints, such as the aggregated stress constraint.

In TO-Master, residual assembly, constitutive evaluation, state solution, objective evaluation, and constraint evaluation are implemented through JAX-FEM[10]. The adjoint method is implemented with customized differentiation rules, which enable the automated computation of the required sensitivity terms. This avoids manual implementation of case-specific sensitivity expressions and keeps the sensitivities consistent with the discretized FEM model.

With the objective and constraint sensitivities available, the design variables are updated using the method of moving asymptotes (MMA)[36]. At each iteration, the FEM solver evaluates the state field, objective, constraints, and gradients, and MMA updates the design variables under the prescribed move limit and constraint settings.

LLM-agent framework design

In this section, we present the LLM-agent framework design of TO-Master. The numerical optimization follows the sensitivity analysis and MMA-based update scheme described in Section "Sensitivity analysis and gradient-based optimization". Built on the agent execution infrastructure provided by Bohrium[37], TO-Master connects user-level problem descriptions with finite-element-based TO tools. As shown in Figure 2, the framework accepts natural language specifications together with optional mesh, geometry, or image inputs. These inputs are translated into a finite element TO model, whose mesh and boundary-condition setup are checked before optimization. The framework then executes the corresponding TO workflow and returns optimized results, field distributions, and convergence histories.

The software stack is organized as follows. Google Agent Development Kit (ADK)[38] runs the LLM agent and manages dialogue state, instructions, and tool selection. The Model Context Protocol (MCP)[39] defines the typed interface through which the agent calls computational tools. The bohr-agent-sdk handles tool execution, status tracking, logging, and artifact management. The TO-Master MCP server implements the domain-specific TO tools, including mesh processing, boundary-condition checking, topology optimization, and result post-processing. The FEA and sensitivity evaluation are performed by the JAX-FEM-based solver described in Section "Sensitivity analysis and gradient-based optimization".

As an LLM-agent framework, TO-Master converts user intent into structured tool arguments and coordinates the execution of the TO workflow. Its numerical reliability comes from coupling this agent-level coordination with deterministic computational tools: mesh generation, mesh inspection, boundary-condition visualization, FEA, sensitivity evaluation, optimization, and post-processing are performed by predefined solver modules. Thus, TO-Master provides a conversational interface for TO problem formulation and execution while preserving the underlying numerical modeling and optimization procedures.

The original evaluations in Section "RESULTS AND DISCUSSION" used DeepSeek-V4-Flash for the benchmark cases, extended cases, and prompt-ablation study. No manual sampling parameters were specified for the main agent; the provider-default inference settings were therefore used. The configuration of the supplementary repeatability experiments is described in Supplementary Material A.

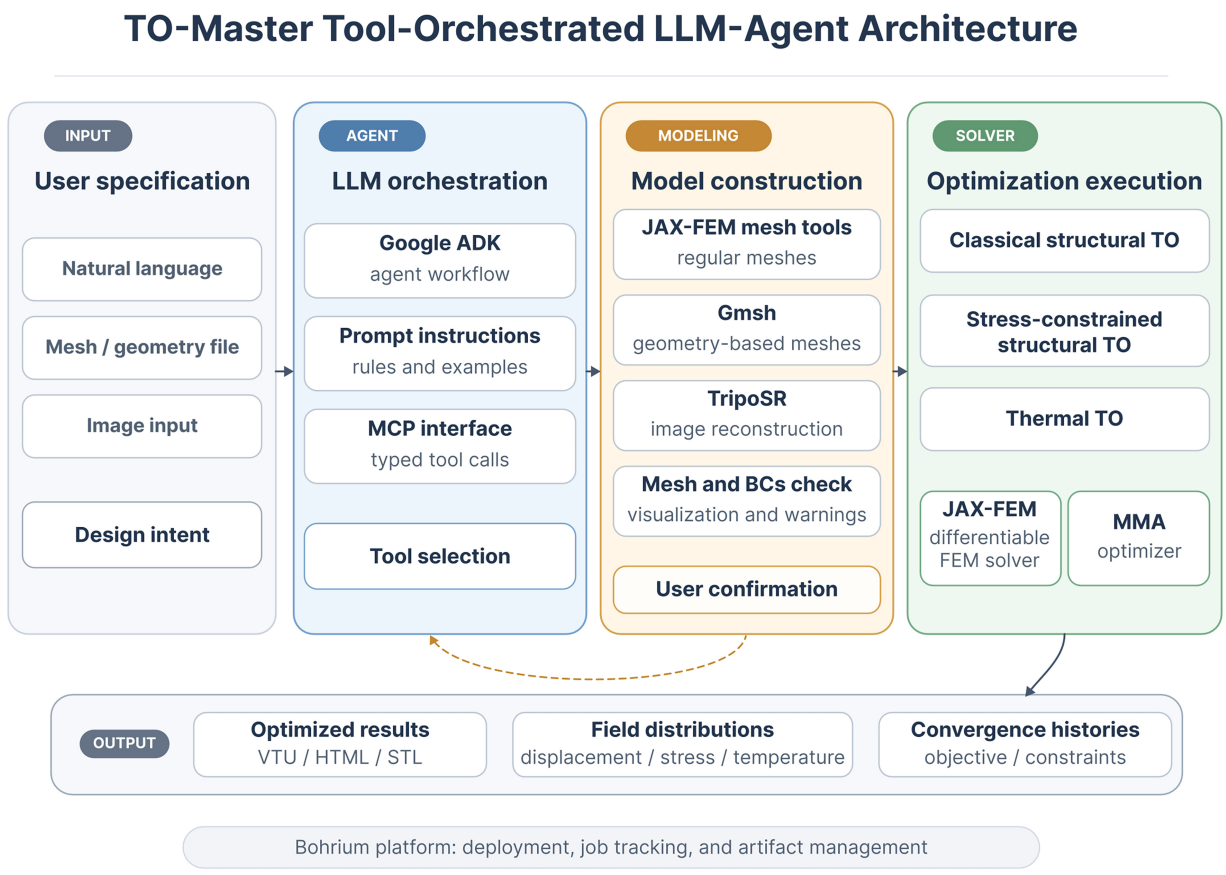


Figure 2. Overview of the TO-Master tool-orchestrated LLM-agent architecture. Natural language specifications and optional mesh, geometry, or image inputs are interpreted by the LLM agent and converted into verified finite element TO models. The framework supports model construction, mesh and boundary-condition checking, optimization execution, and the generation of optimized results, field distributions, and convergence histories. ADK: Agent Development Kit; BCs: boundary conditions; FEA: finite element analysis; FEM: finite element method; HTML: HyperText Markup Language; LLM: large language model; MCP: Model Context Protocol; MMA: method of moving asymptotes; STL: stereolithography; TO: topology optimization; VTU: Visualization Toolkit XML-based unstructured-grid file format.

The following subsections describe the main framework components of TO-Master: the few-shot and reasoning-guided prompt design, the TO-Master tool layer, and the LLM-agent-orchestrated workflow.

Few-shot and reasoning-guided prompt design

The TO-Master agent is configured by a structured system instruction, whose organization is shown in Figure 3. The complete instruction used in the numerical experiments is provided in Supplementary Material B. The instruction contains four parts: role specification, tool-usage rules, internal reasoning guidance, and few-shot examples.

The role specification defines the agent as a domain-specialized assistant for JAX-FEM mesh generation, mesh and boundary-condition checking, and topology optimization. This role definition anchors the agent to TO-related tasks and prevents it from treating the interaction as a general code-generation problem.

The tool-usage rules define the allowed computational interface. They specify the available MCP tools, legal boundary-condition selection formats, expected argument fields, returned artifact structures, and user-confirmation requirements. In particular, topology optimization tools require prior mesh and

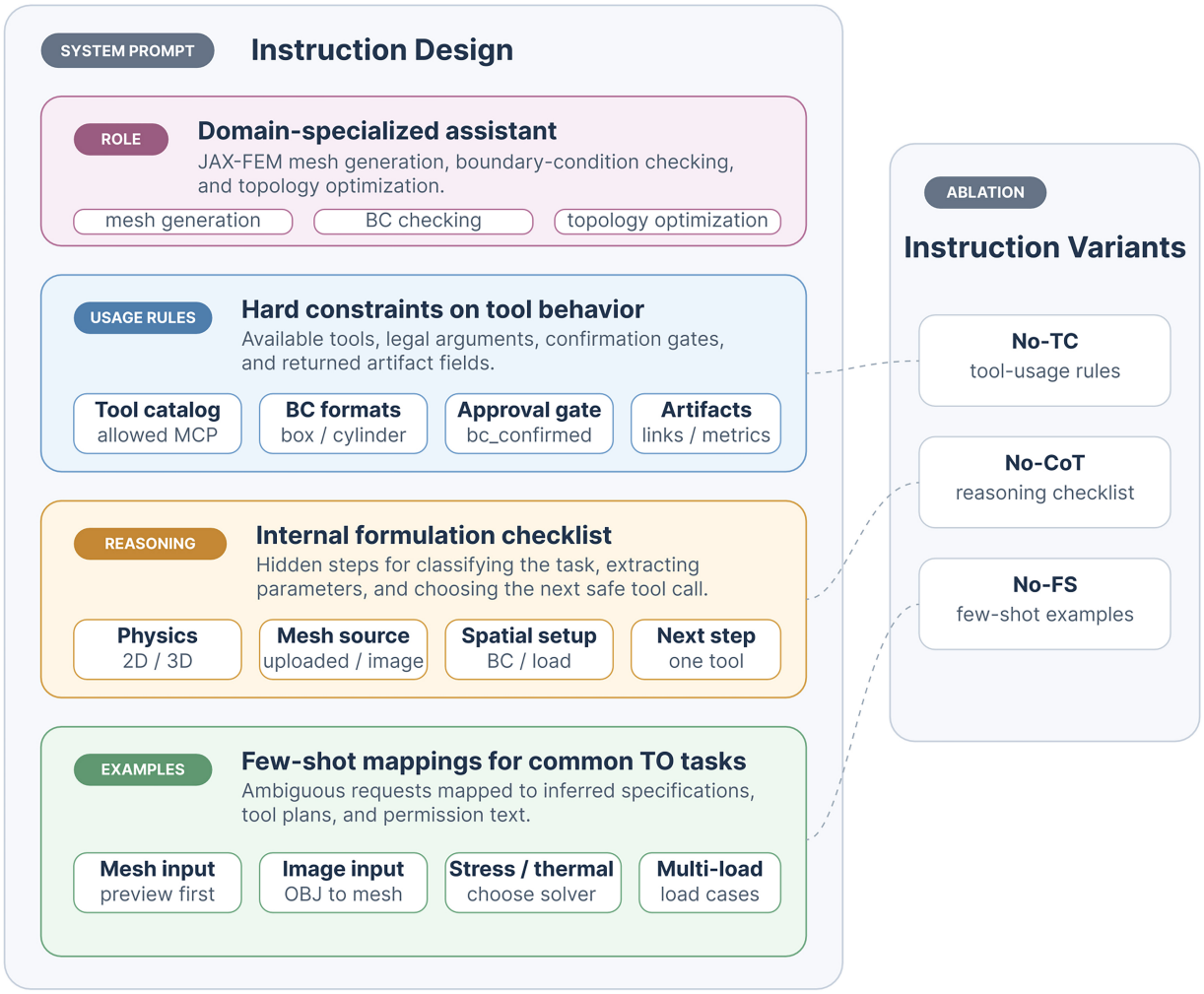


Figure 3. Structure of the TO-Master system instruction. The instruction comprises role specification, tool-usage rules, internal reasoning guidance, and few-shot examples. The ablation study separately removes the latter three components, while role specification is retained in all configurations. 2D: Two-dimensional; 3D: three-dimensional; BC: boundary condition; MCP: Model Context Protocol; No-CoT: removal of chain-of-thought guidance; No-FS: removal of few-shot examples; No-TC: removal of tool-interface documentation; OBJ: Wavefront OBJ geometry-file format; TO: topology optimization.

boundary-condition checking, followed by user confirmation through the bc_confirmed flag. These rules constrain the execution order and prevent the agent from directly launching an optimization run before the finite element TO model has been checked.

The internal reasoning guidance provides a formulation checklist for the agent. It guides the agent to identify the problem type, distinguish mechanical and thermal TO, determine the spatial dimension, identify the mesh source, extract boundary conditions, loads, passive domains, material parameters, volume fraction, and optimization settings, and then select the next safe tool call. This reasoning process is used internally and is not shown to the user; the user only receives the resulting permission request, tool output summary, and links to generated artifacts.

The few-shot examples provide concrete mappings from ambiguous user requests to complete tool plans. They cover uploaded mesh preview, image-to-mesh conversion, mechanical compliance TO, stress-constrained TO, thermal TO, multi-load-case TO, and unsupported direct-solve requests. Each example includes an easy prompt, an inferred complete specification, a concrete tool plan, and a user-visible

permission request. These examples help the agent resolve common ambiguities, such as finite load patches, passive-domain assignment, insulated thermal boundaries, and the choice between single-load, multi-load, and stress-constrained solvers.

TO-Master tool layer

The TO-Master MCP server exposes domain-specific tools for mesh preparation, mesh and boundary-condition checking, TO execution, and postprocessing. Each tool is called with typed arguments and returns structured outputs, including execution status, input parameters, artifacts, metrics, and text summaries. This tool layer is central to TO-Master: it allows the LLM agent to automate TO workflows by orchestrating deterministic numerical modules, rather than by generating solver code.

For mesh preparation, TO-Master supports generated meshes, uploaded meshes, and image-derived mesh models. Rectangular and box meshes are generated using JAX-FEM-based utilities, while cylinder and geometry-based box meshes can be generated using Gmsh[40]. Uploaded meshes are inspected before use, with dimensions, bounding boxes, node and cell counts, and preview images reported to the user. For image inputs, a TripoSR-based module[41] reconstructs an OBJ surface, which is then converted into a voxelized hexahedral volume mesh. This enables the same agent workflow to handle regular benchmark domains, user-provided engineering meshes, and image-derived geometry inputs.

For mesh and boundary-condition checking, TO-Master visualizes supports, loads, and thermal boundary conditions selected by box or cylindrical regions. The checking tools report empty selections and return preview images and interactive HyperText Markup Language (HTML) files. For mechanical problems, a full-material FEA check is also used to detect nearly trivial displacement responses, which may indicate ineffective loads or ill-posed boundary conditions. For problems with multiple load cases, each load case is checked separately.

With prior checking and user confirmation through the bc_confirmed flag, which reduces the risk of silent formulation errors, the optimization tools support 2D and 3D structural compliance TO, multi-load-case structural compliance TO, stress-constrained structural TO, and steady-state thermal conduction TO. They share common arguments for mesh paths, boundary conditions, material parameters, volume fractions, filter settings, move limits, iteration numbers, optional Heaviside projection, and passive domains. Passive domains are passed as undesign_domains and kept solid during optimization. Postprocessing exports optimized density fields, field distributions, convergence histories, Visualization Toolkit XML-based unstructured-grid (VTU) files, interactive HTML views, and stereolithography (STL) surface models for optimized results, making the optimized results and convergence behavior inspectable.

LLM-agent-orchestrated workflow

The agent workflow converts a user request into a checked and executable TO problem. The LLM agent interprets the design domain, mesh source, boundary conditions, loads, passive domains, material parameters, objective type, physical constraints, and optimization settings, and then selects the appropriate mesh, checking, and optimization tools. Regular domains are generated directly, uploaded meshes are previewed before use, and image inputs are processed through the image-to-geometry and geometry-to-mesh pipeline.

A key feature is staged user confirmation. Before critical tool calls, the agent summarizes the inferred action, names the tool to be called, lists the key parameters, and asks for user approval. After mesh and boundary-condition checking, the agent reports artifacts, metrics, and warnings, and proceeds to optimization only after confirmation. If a tool returns a warning or failure status, the agent reports the issue and pauses instead of continuing silently.

After confirmation, the selected optimization tool is called with typed arguments. The deterministic solver evaluates the finite element state, objective, constraints, and sensitivities, and MMA updates the density variables as described in Section "Sensitivity analysis and gradient-based optimization". After optimization, TO-Master returns optimized results, field distributions, convergence histories, and interactive visualizations. In this workflow, the LLM handles interpretation, tool selection, and user interaction, while deterministic tools handle mesh processing, FEA, optimization, and postprocessing. This division makes the workflow reproducible while retaining the flexibility of natural language interaction.

RESULTS AND DISCUSSION

Benchmark and engineering examples

This section evaluates the robustness and applicability of TO-Master through a set of benchmark and engineering topology optimization examples. The examples cover two- and three-dimensional benchmark structures, thermal conduction, multi-load structural compliance minimization, stress-constrained problems, arbitrary geometries, and image-to-optimization cases. They are designed to assess whether TO-Master can translate natural language specifications into solver-ready topology optimization problems across different problem settings.

All examples use the same differentiable FEM backend and gradient-based optimization framework. The geometry, boundary conditions, and optimization constraints are varied across cases. For each example, the constructed boundary conditions, convergence histories, constraint values, and optimized results are recorded for analysis.

The problem specifications for all eleven cases were prepared by the first author before being provided to TO-Master. These specifications serve as the expert reference for evaluating the agent-generated optimization inputs. For Cases 1-6, the precise-prompt reference runs from the ablation study are used consistently in Figures 4 and 5.

For the structural cases with regular geometries (Cases 1-4 and 6-8), the adopted unit system is m-N-Pa: coordinates and displacements are expressed in m, forces in N, and Young's modulus, surface tractions, and stresses in Pa. Cases 9 and 10 use the mm-μN-Pa convention described in Section "Optimization for structures with arbitrary geometries", while Case 11 uses normalized geometry coordinates. For the thermal case, lengths are expressed in m, temperature in °C, thermal conductivity in W/(m·K), and the volumetric heat source in W/m^3, with a unit-thickness interpretation of the 2D domain. The solver uses numerical inputs directly without automatic unit conversion. Mechanical loads described as "pressure" in the prompts are implemented as prescribed traction vectors.

All coordinate bounds and traction components are specified in the coordinate system of the corresponding mesh. The convention in Supplementary Material B maps "downward" to negative y in 2D and negative z in 3D when no coordinate direction is explicitly specified. Explicit coordinate directions take precedence; Cases 7, 8, and 10 prescribe loading in the negative y-direction of their respective mesh coordinate systems.

The eleven cases are organized into regular benchmark cases and extended engineering examples. Cases 1-6 test standard structural, thermal, and multi-load formulations on regular domains, whereas Cases 7-11 examine stress constraints, arbitrary engineering geometries, and image-to-optimization.

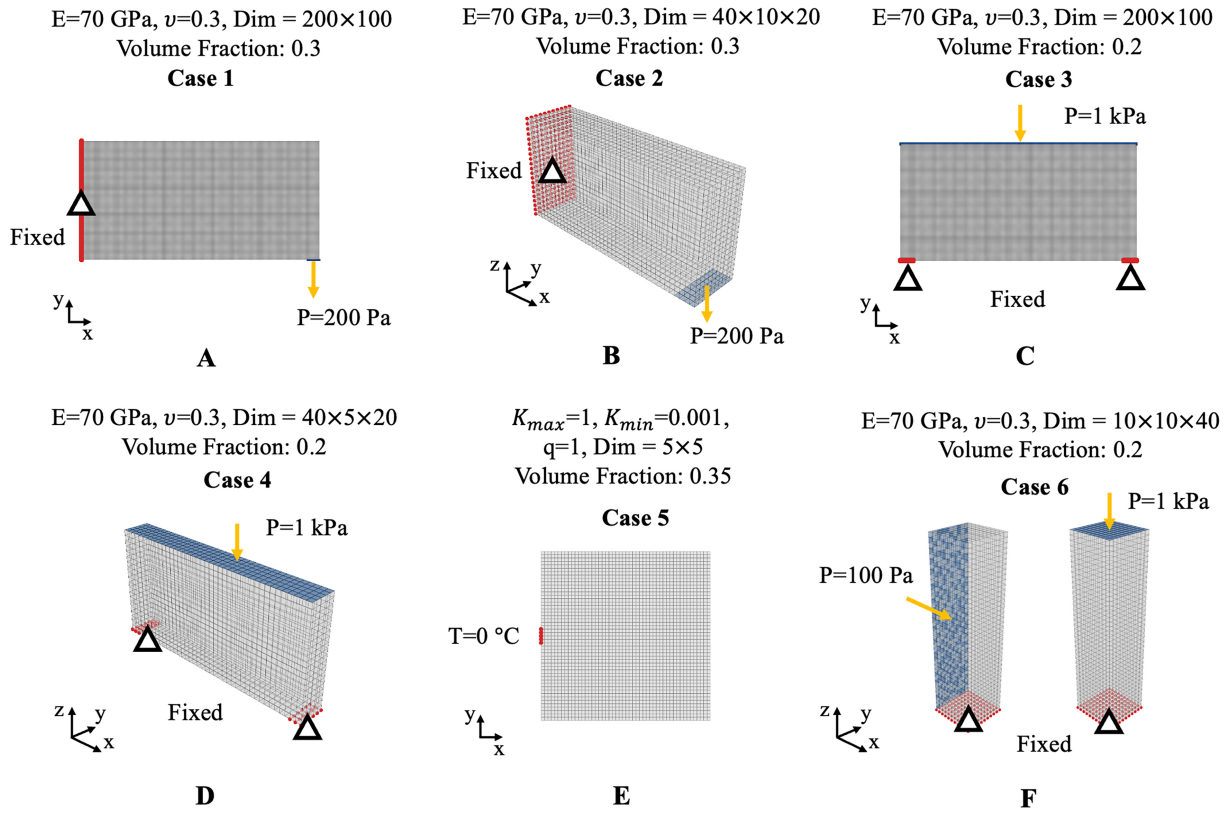


Figure 4. Boundary conditions for the benchmark cases. (A) Case 1, 2D cantilever beam; (B) Case 2, 3D cantilever beam; (C) Case 3, 2D bridge; (D) Case 4, 3D bridge; (E) Case 5, 2D heat conduction; (F) Case 6, 3D tower with multiple load cases. 2D: Two-dimensional; 3D: three-dimensional.

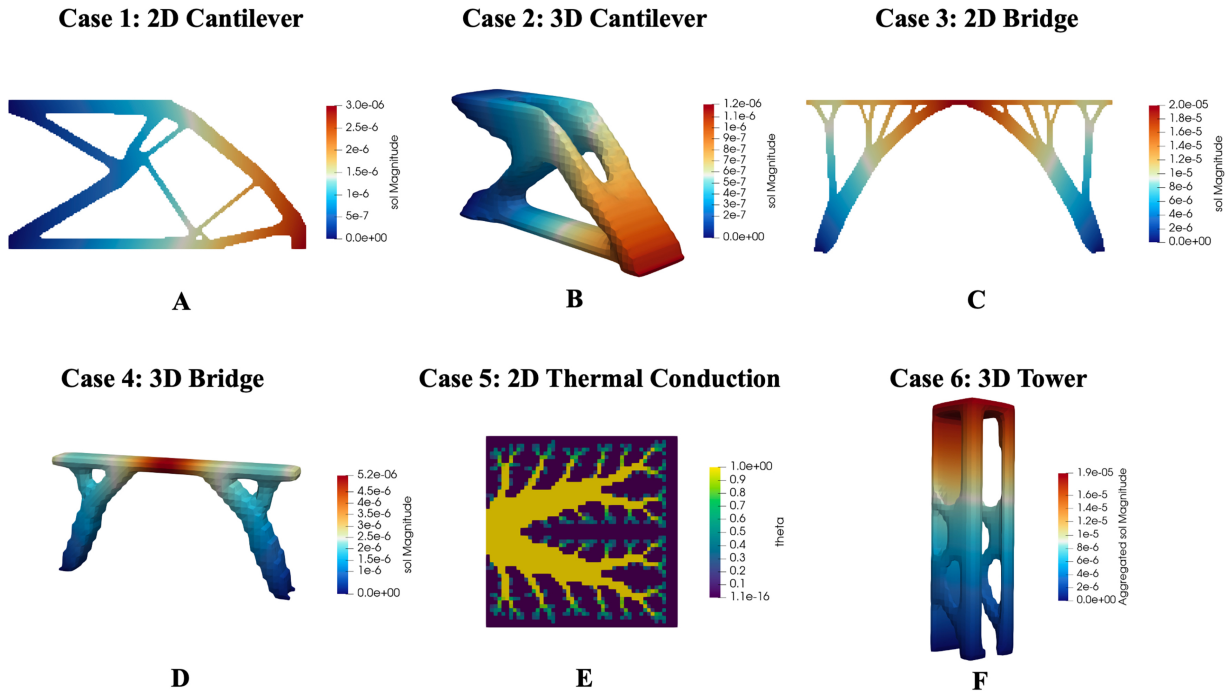


Figure 5. Topology optimization results for the benchmark cases. The figure depicts the displacement magnitude for Cases 1-4, the material density field for Case 5, and the aggregated displacement magnitude for Case 6. (A) Case 1, 2D cantilever beam; (B) Case 2, 3D cantilever beam; (C) Case 3, 2D bridge; (D) Case 4, 3D bridge; (E) Case 5, 2D heat conduction; (F) Case 6, 3D tower with multiple load cases. For Case 6, the aggregated displacement magnitude is defined as $u_{agg} = \sqrt{\left|\mathbf{u}^{(1)}\right|_2^2 + \left|\mathbf{u}^{(2)}\right|_2^2}$, where $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ are the displacement fields obtained from the two separate load-case solutions. 2D: Two-dimensional; 3D: three-dimensional.

Table 2. Summary of the benchmark and extended topology optimization cases evaluated using TO-Master

Case ID	Geometry	Boundary conditions
1	2D cantilever beam	Left edge fixed; surface traction applied at the lower-right corner
2	3D cantilever beam	Left side fixed; surface traction applied at the lower-right corner
3	2D bridge	Bottom-left and bottom-right corners fixed; surface traction applied on the top boundary
4	3D bridge	Bottom-left and bottom-right corners fixed; surface traction applied on the top boundary
5	2D thermal conduction	Uniform internal heat source; fixed-temperature cold sink applied to the central region of the left boundary
6	3D tower (Multiple load cases)	Bottom fixed; two load cases considered: (1) downward surface traction applied on the top surface; (2) rightward surface traction applied on the left surface
7	3D L-Shaped beam (Stress constraint)	Top-left support region fixed; surface traction in the negative *y*-direction applied on the right-side loading patch
8	3D T-Shaped beam (Stress constraint)	Two support regions fixed; surface traction in the negative *y*-direction applied on the central loading patch
9	3D Jet engine bracket (Arbitrary geometry)	Four lower bolt holes fixed; inclined surface traction applied at the upper pin hole
10	3D Airplane bearing bracket (Arbitrary geometry)	Four lower bolt holes fixed; surface traction in the negative *y*-direction applied at the right-side pin hole
11	3D Dog-Bone-Shaped specimen (Image input)	Left end fixed; downward surface traction applied at the right end

2D: Two-dimensional; 3D: three-dimensional.

Table 2 lists the eleven cases with their geometries and boundary conditions. Cases 1-6 are regular geometry problems, including two-dimensional and three-dimensional structural compliance minimization, thermal conduction, and a multi-load tower structure. Cases 7-8 are stress-constrained optimization problems. Cases 9-10 involve arbitrary three-dimensional engineering geometries. Case 11 uses a geometry and mesh reconstructed from a 2D image input.

All cases are formulated through natural language interaction with TO-Master, without user-written solver code. The detailed prompts are provided in Supplementary Material C. The corresponding boundary conditions are shown in Figures 4 and 6, and the optimized results are shown in Figures 5 and 7.

Optimization on regular benchmark problems

Cases 1-6 in Table 2 evaluate the basic optimization capability of TO-Master on regular domains. Their boundary conditions are shown in Figure 4, and the corresponding optimized results are shown in Figure 5. These cases include cantilever beams, bridge structures, a thermal conduction problem, and a multi-load tower problem.

The thermal benchmark, Case 5, uses a square domain with dimensions 5 m × 5 m and an element size of 0.1 m. A uniform internal heat source of 1 $W \cdot m^{-3}$ is applied over the whole domain. The cold sink is prescribed as a reference temperature $T = 0$ °C on the middle part of the left boundary. The maximum and minimum thermal conductivities are $k_{max} = 1.0$ $W \cdot m^{-1} \cdot K^{-1}$ and $k_{min} = 10^{-3}$ $W \cdot m^{-1} \cdot K^{-1}$, respectively. The other boundaries are insulated by omitting heat-flux boundary conditions.

For clarity, the intermediate interaction process is illustrated using the three-dimensional bridge case as a representative example (Case 4 in Table 2; see Figure 4D and Figure 5D). The design domain has dimensions 40 m × 5 m × 20 m. The lower-left and lower-right corner regions are fixed with zero displacement, and a downward surface traction of 1,000 Pa is applied on the top surface. The top layer is preserved as a passive

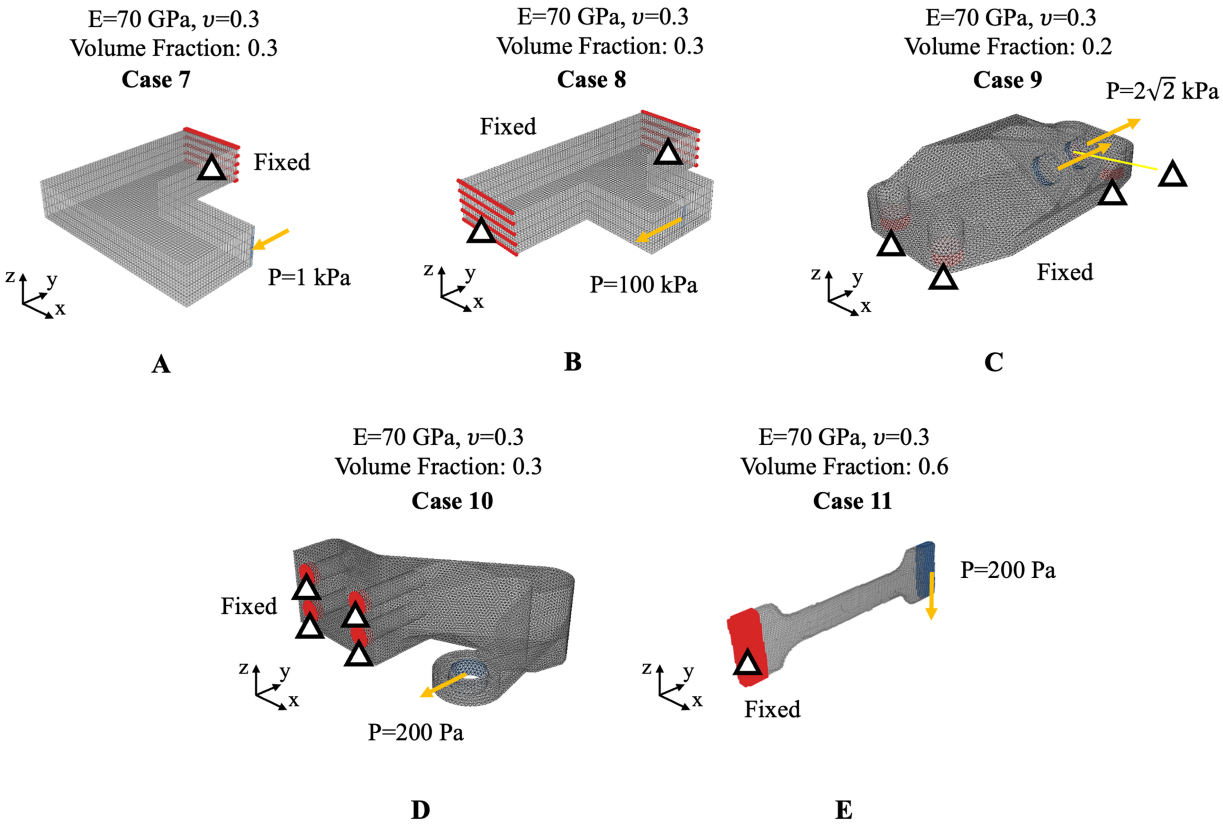


Figure 6. Boundary conditions for the extended cases. (A) Case 7, 3D L-shaped beam with stress constraint; (B) Case 8, 3D T-shaped beam with stress constraint; (C) Case 9, 3D jet engine bracket with arbitrary geometry; (D) Case 10, 3D airplane bearing bracket with arbitrary geometry; (E) Case 11, 3D dog-bone-shaped specimen generated from image input. Geometry coordinates are expressed in mm for Cases 9-10 and in normalized length units for Case 11. 3D: Three-dimensional.

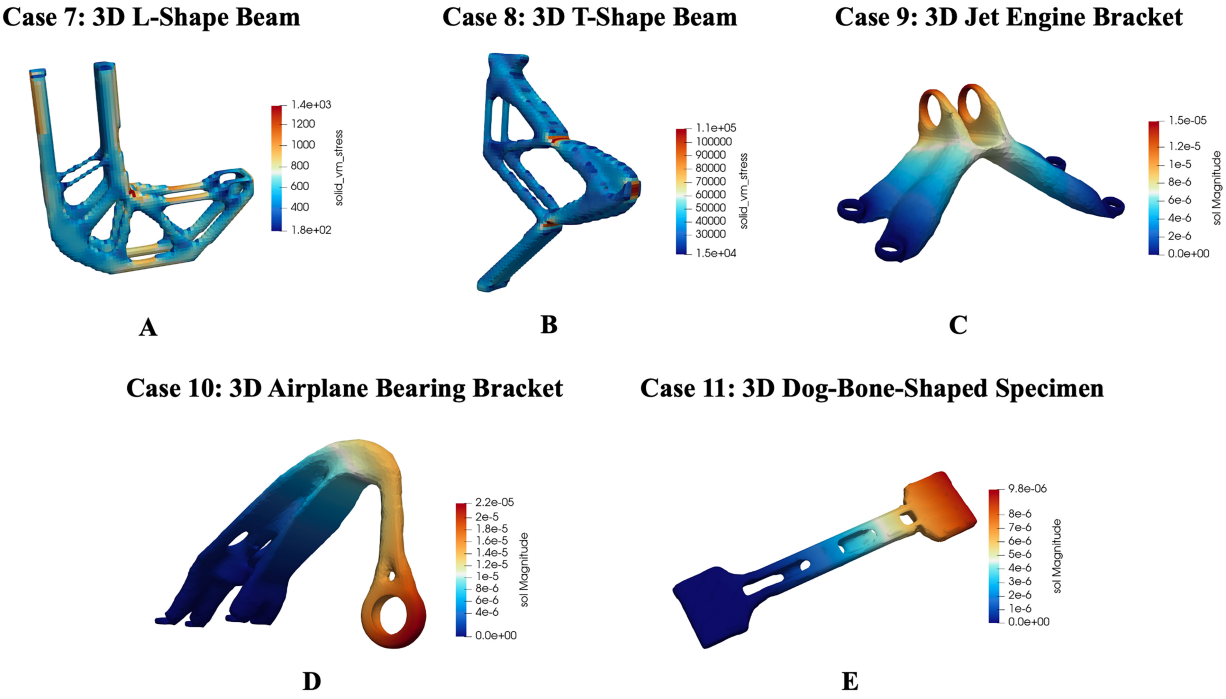


Figure 7. Topology optimization results for the extended cases. For each case, the figure depicts the von Mises stress or the displacement magnitude of the optimized results under the prescribed boundary conditions. (A) Case 7, 3D L-shaped beam with stress constraint; (B) Case 8, 3D T-shaped beam with stress constraint; (C) Case 9, 3D jet engine bracket with arbitrary geometry; (D) Case 10, 3D airplane bearing bracket with arbitrary geometry; (E) Case 11, 3D dog-bone-shaped specimen generated from image input. For Cases 7 and 8, the displayed stress field is evaluated on the thresholded solid topology. 3D: Three-dimensional.

domain to maintain the loading interface during optimization. The default Young's modulus is $E = 70$ GPa, the Poisson's ratio is $\nu = 0.3$, and a volume fraction constraint of 0.2 is imposed. The optimization is run for 100 iterations.

A fuzzy natural language prompt used for this case is shown below. This prompt is the same as the fuzzy prompt used in the ablation study. The corresponding coordinate-level precise prompt is provided in Supplementary Material C.

· Fuzzy natural language problem description used for Case 4

Create a 3D bridge benchmark with a 40 × 5 × 20 mesh. Support both lower-end corners and apply downward pressure of 1,000 across the top surface. Keep the top layer as a passive domain. Use 20 percent material, run 100 optimization steps, set the step size to 0.1, and use a filter factor of 2.

Using the full instruction, TO-Master translates the fuzzy natural language description into a solver-ready problem, including mesh generation, boundary condition visualization, passive-domain assignment, and topology optimization.

The optimized results exhibit load paths and structural features qualitatively consistent with classical topology optimization benchmarks.

Stress-constrained optimization

Cases 7 and 8 evaluate the ability of TO-Master to construct stress-constrained topology optimization problems from natural language descriptions. The boundary conditions and optimized results are shown in Figure 6A and B and Figure 7A and B, respectively. The corresponding iteration histories are reported in Figure 8.

Unlike standard structural compliance minimization, these cases require the agent to parse the stress limit and stress-related solver parameters in addition to the geometry, supports, and loads.

In Case 7, the L-shaped beam is fixed at the top-left support region. A surface traction of -1,000 Pa in the y-direction is applied to the right-side loading patch, which is preserved as a passive domain. The optimization is performed with a volume fraction of 0.3, 150 optimization iterations, a filter factor of 2, and a maximum allowable density-weighted von Mises stress of 1,360 Pa.

In Case 8, the T-shaped beam is fixed at two support regions. A surface traction of -100,000 Pa in the y-direction is applied to the central loading patch, which is also assigned as a passive domain. The optimization uses a volume fraction of 0.3, 150 optimization iterations, a filter factor of 2, and a maximum allowable density-weighted von Mises stress of 111,500 Pa.

The prescribed stress limits of 1,360 Pa and 111,500 Pa for Cases 7 and 8, respectively, were selected through preliminary numerical trials as parameters for these demonstration cases.

The final maximum density-weighted von Mises stresses are 1,359.75 Pa for Case 7 and 110,687.88 Pa for Case 8, both below their prescribed limits. After thresholding the physical density at 0.5 and re-evaluating the resulting solid topology, the maximum von Mises stresses over the solid elements are 1,380.93 Pa and 110,413.67 Pa, respectively. The post-processed result for Case 7 therefore exceeds its prescribed limit by approximately 1.54%, whereas Case 8 remains below its limit. These results distinguish the density-weighted stress measure used during optimization from the solid-material stress evaluated after thresholding.

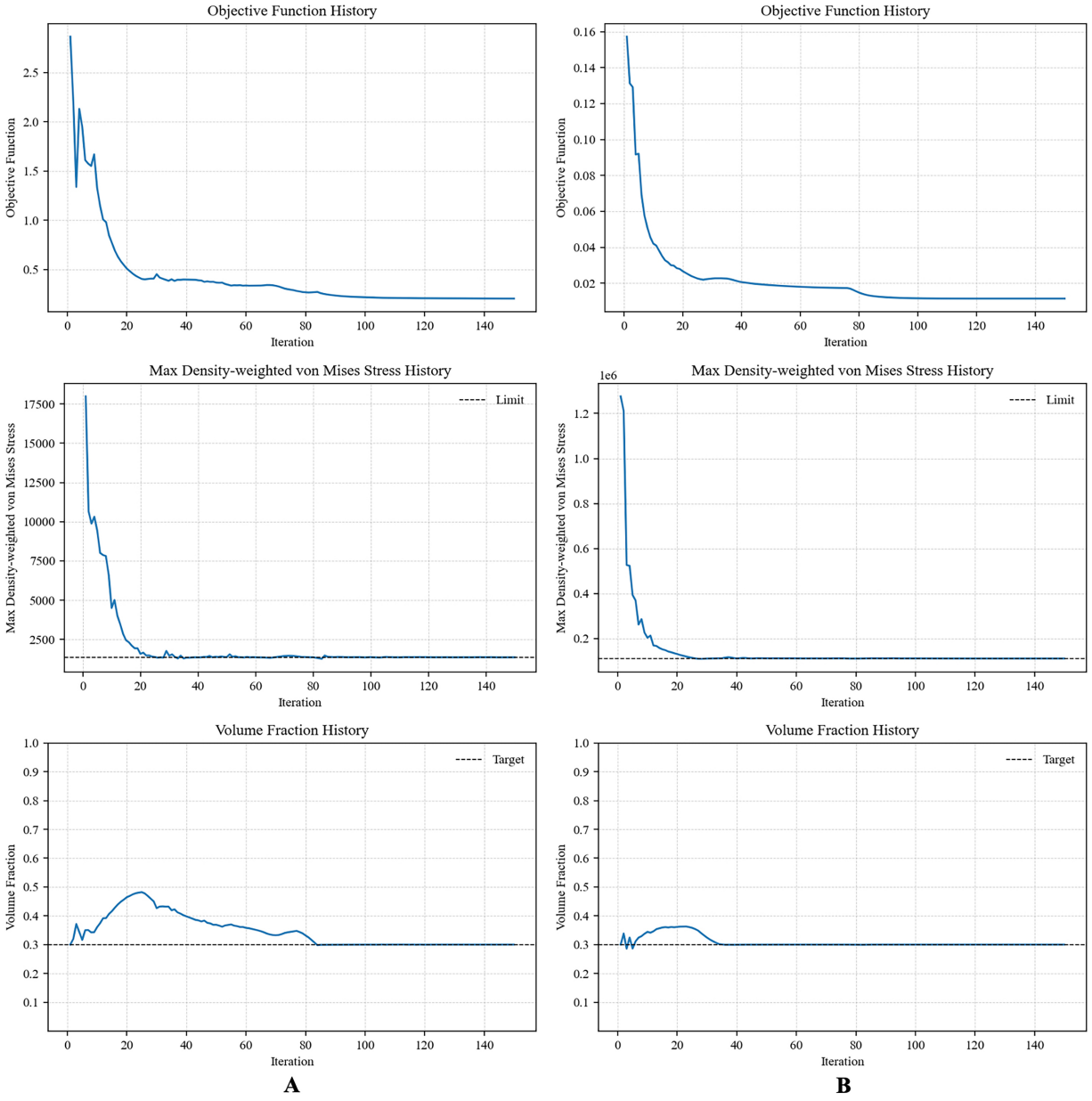


Figure 8. Iteration histories for the two stress-constrained cases. For each case, the figure depicts the histories of the objective function, the maximum density-weighted von Mises stress, and the volume fraction. (A) Case 7, 3D L-shaped beam; (B) Case 8, 3D T-shaped beam. 3D: Three-dimensional.

Optimization for structures with arbitrary geometries

Cases 9 and 10 examine TO-Master on arbitrary engineering geometries. The geometries of the jet engine bracket (Case 9) and airplane bearing bracket (Case 10) originate from the GE Jet Engine Bracket Challenge[42] and the Alcoa Airplane Bearing Bracket Challenge[43], respectively, both hosted on GrabCAD. Their boundary conditions and optimized results are shown in Figure 6C and D and Figure 7C and D, respectively. Compared with regular box-like domains, these cases require the agent to associate supports, loads, and passive domains with cylindrical geometric features such as bolt holes and pin holes.

Cases 9 and 10 use a consistent mm-μN-Pa unit system. Mesh coordinates and displacements are expressed in mm, while Young's modulus, surface tractions, and stresses are expressed in Pa (equivalently, μN/mm^2). Compliance is expressed in μN·mm.

In Case 9, the jet engine bracket is fixed at four lower cylindrical bolt-hole regions. A combined surface traction with components of 2,000 Pa in the x- and z-directions, corresponding to a resultant magnitude of $2\sqrt{2}$ kPa, is applied at the upper pin-hole cylinder. The fixed bolt-hole regions and the loaded pin-hole region are preserved as passive domains. The case is solved as a structural compliance minimization problem with a volume fraction of 0.2 and 150 optimization steps.

In Case 10, the airplane bearing bracket is fixed at four cylindrical mounting holes. A surface traction of -200 Pa in the y-direction of the mesh coordinate system is applied to the right-side cylindrical pin region. The mounting holes and loading cylinder are preserved as passive domains. The optimization is performed with a volume fraction of 0.3 and 150 steps.

These two cases demonstrate the execution of the TO-Master workflow on engineering geometries with nontrivial boundary conditions and tetrahedral unstructured meshes.

Image to optimization

Case 11 evaluates the image-to-optimization workflow. The boundary conditions and optimized results are shown in Figure 6E and Figure 7E, respectively. For the image-derived case, the geometry and all spatial coordinates are expressed in normalized length units. No physical length calibration is applied to the reconstructed geometry in this example. Physical coordinates can be obtained by assigning a reference length to one normalized coordinate unit; computed displacements share the same length scale.

The reconstructed optimization mesh contains 28,738 nodes and 23,345 eight-node hexahedral (HEX8) elements. The numerical selections use $x \in [-0.035, 0.042]$ and $z \in [-0.2, 0.2]$. Within these bounds, the region $y \in [-0.7, -0.4]$ is fully fixed, and a surface traction of (0, 0, -200) Pa is applied to the exterior boundary within $y \in [0.4, 0.7]$. The regions $y \in [-0.7, -0.3]$ and $y \in [0.3, 0.7]$ are retained as solid passive domains. Compliance minimization is performed with a prescribed design-domain volume fraction of 0.6 for 50 iterations.

This case shows that TO-Master can connect image-based geometry generation with topology optimization. The optimized result illustrates the application of the topology optimization workflow to the reconstructed geometry under the specified support and loading conditions.

For the eleven cases presented above, we compared the agent-generated optimization inputs with the expert-prepared "Clear Problem Specification" prompts in Supplementary Material C, using the same numerical backend and default parameters. As summarized in Supplementary Table 1 [Supplementary Material D], the inputs matched the expert settings in Cases 1, 2, and 5-10. In Cases 3 and 4, some coordinate ranges used to select supports, loads, or passive domains were expanded. In Case 4, the load-selection box was also restricted to the mesh extents without changing the selected loaded faces. In Case 11, the unspecified x-range was completed using rounded mesh bounds, while the prescribed y- and z-ranges were retained. Overall, TO-Master preserved the prescribed traction or thermal parameters, volume fractions, iteration counts, and applicable stress limits across all eleven cases, with the identified differences confined to spatial region selection.

The mesh sizes, final design-domain volume fractions, compliance values, and recorded execution times for Cases 7-11 are summarized in Supplementary Table 2 [Supplementary Material A]. Final volume fractions were close to their prescribed values. In five independent agent sessions per case, the requested workflow was completed in all five runs for Cases 8 and 9 and in four runs for Case 11 [Supplementary Table 3, Supplementary Material A]. Cases 8 and 9 each yielded identical reported compliance values and volume

fractions across the five runs. Among the four completed Case 11 runs, three produced identical results, with final compliance 0.053% higher than the original result obtained on macOS. One completed run used a slightly smaller loading region and produced a different compliance value, as detailed in Supplementary Material A. Overall, most repeated runs produced similar optimized structures and numerical results, supporting the repeatability of TO-Master for the selected cases.

Ablation study on prompt design and problem specifications

This section evaluates the effect of instruction design under ambiguous user input. The ablation study is conducted on the six regular benchmark cases shown in Figure 4, i.e., Cases 1-6 in Table 2. The corresponding reference optimization results are shown in Figure 5.

For each case, the result obtained using the full instruction and the precise problem description is treated as the reference solution. Then, a fuzzy problem description is tested under four instruction configurations: the full system instruction (full prompt, FP), removal of few-shot examples (No-FS), removal of chain-of-thought guidance (No-CoT), and removal of tool-interface documentation (No-TC). These variants correspond to removing the few-shot examples, internal reasoning guidance, and tool-usage rules from the prompt design described in Section "LLM-agent framework design" and Supplementary Material B. Each fuzzy-prompt condition was repeated five times. In total, the ablation study contains 6 reference runs and $6 \times 4 \times 5 = 120$ fuzzy-prompt runs.

In the No-TC configuration, the USAGE_RULES section is removed from the system instruction, while the same MCP toolset remains available. The agent therefore retains access to the tool names, descriptions, and declared parameter names and types exposed through the MCP tool schemas. The removed section provides additional interface documentation, including the tool catalog, boundary-condition formats, argument conventions, and result-field descriptions, together with tool-use and confirmation rules. No-TC thus evaluates removal of this prompt-level interface documentation and its associated usage guidance, rather than removal of the tools or their schemas.

The instruction contains examples and inference rules tailored to the benchmark problems evaluated here. In particular, the thermal middle-sink example in Supplementary Material B closely matches Case 5. The formulation checklist also includes benchmark-specific guidance on lower-corner bridge supports, top passive regions, and a cold sink spanning the central 10% of the left boundary. The ablation study therefore evaluates the contribution of instruction components on problems around which the instruction was designed, rather than demonstrating generalization to unseen problems.

The evaluation follows a two-stage protocol. First, the boundary condition checker is used to detect invalid or suspicious boundary and load definitions. A run is counted as failed if the checker reports a warning or timeout. Second, if the solver-ready optimization parameters match a previous run for the same case, the optimization is not repeated, and the run inherits the outcome of the matched case. Otherwise, the optimization is executed, and the optimized result is reviewed against the reference solution. A run is considered successful if expert review confirms that the boundary conditions and passive domains are correctly constructed, the prescribed constraints are satisfied, and the optimized result is physically reasonable and visually comparable to the reference result. Runs requiring manual assessment were reviewed by the first, second, and fifth authors using the criteria described above. During the ablation runs, requests to proceed were answered automatically by the experimental script.

The aggregated success rates are summarized in Table 3 and Figure 9, while representative failure cases are shown in Figure 10. The 95% confidence intervals are Wilson score intervals for the pooled success proportions, with 30 trials per configuration. The full instruction achieves the highest success rate, with only

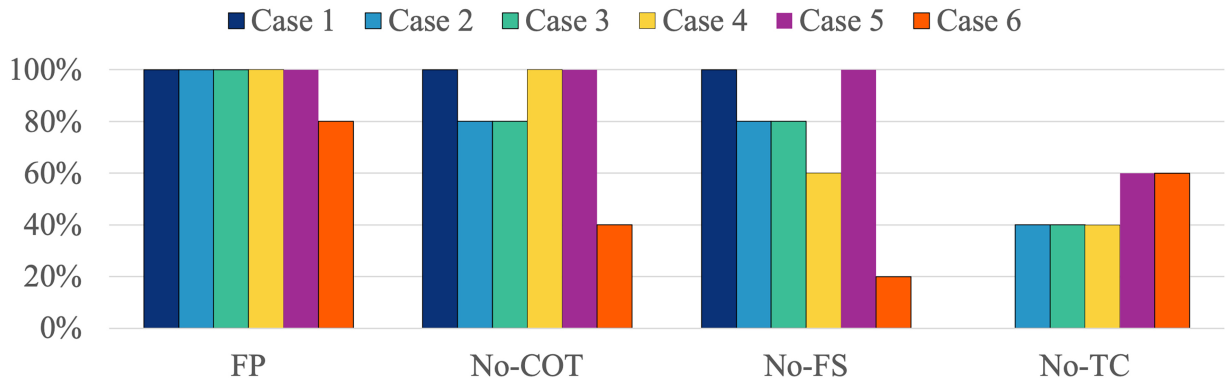


Figure 9. Success rates of TO-Master under different instruction configurations with fuzzy problem specifications. FP: Full prompt (full system instruction); No-CoT: removal of chain-of-thought guidance; No-FS: removal of few-shot examples; No-TC: removal of tool-interface documentation.

one failed trial caused by passive-domain misinterpretation in the multi-load tower case. This indicates that the complete instruction design can usually resolve ambiguity in fuzzy user prompts.

Removing tool-interface documentation causes the largest performance drop. Most failures in the No-TC setting are caused by malformed or inconsistent boundary condition definitions, such as repeated constrained components, missed load regions, or invalid displacement values. These errors are detected by the boundary condition checker before expensive optimization is performed.

Removing few-shot examples also reduces robustness. In the No-FS setting, the agent more frequently misinterprets support or load regions, especially in 3D bridge and multi-load cases. Some trials fail before optimization due to boundary condition timeouts, while others proceed to optimization but produce physically inconsistent results.

The No-CoT setting performs better than No-FS and No-TC but still fails in several 3D cases. The dominant failures are passive-domain errors in the multi-load tower and a load-selection mismatch in the 3D cantilever case. FP achieved a higher observed success rate than No-CoT (29/30 versus 25/30), but the difference was not statistically significant in a two-sided Fisher's exact test ($P = 0.195$).

The 3D cantilever trial in Figure 10A illustrates a load-selection error. The No-CoT trial applies a traction of (0, 0, -200) Pa over [38, 40, 0, 3, 0, 0], whereas the expert reference uses [36, 40, 0, 10, 0, 0]. The corresponding loaded areas are 6 and 40 m^2, giving resultant force magnitudes of 1,200 and 8,000 N, respectively. The trial satisfies the volume constraint, with a final volume fraction of 0.299908 against the prescribed limit of 0.3. Its final compliance is 0.000210780 N·m, compared with 0.00906013 N·m for the reference. Because the applied loads differ, the lower compliance does not indicate improved optimization performance. This trial is therefore classified as a formulation failure because it does not reproduce the expert-specified loading.

Overall, FP achieved the highest observed success rate, while No-TC showed the largest decrease. These results suggest that instruction design affects formulation success on the evaluated cases, although the FP-No-CoT comparison does not establish a statistically significant benefit from reasoning guidance.

Discussion

The benchmark and engineering examples show that TO-Master can construct TO formulations from natural language input, including single-load and multi-load structural compliance minimization, as well as

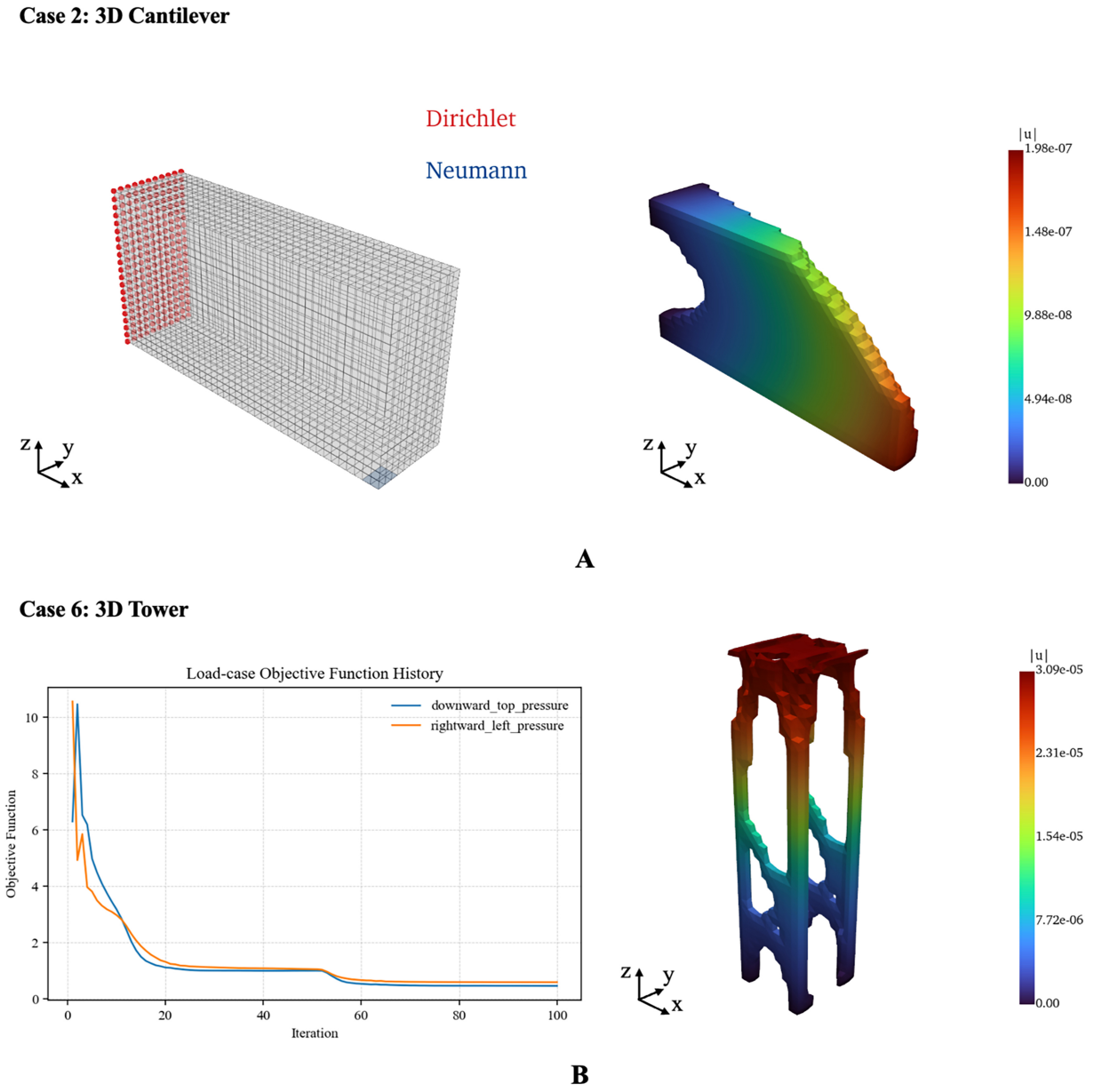


Figure 10. Representative failure cases under fuzzy-prompt ablation settings. In the 3D cantilever case (A), the loaded patch is restricted to one side of the beam, departing from the expert reference loading. In the 3D tower case (B), the passive domain is misinterpreted, causing the loaded surface to be removed during optimization. 3D: Three-dimensional.

Table 3. Aggregated success rates and dominant failure modes of TO-Master under fuzzy problem specifications with different instruction configurations across the six regular benchmark cases

Prompt configuration	Success rate (%)	95% confidence interval (%)	Dominant failure mode
FP	96.7	83.3-99.4	Passive-domain misinterpretation
No-CoT	83.3	66.4-92.7	Passive-domain misinterpretation
No-FS	73.3	55.6-85.8	Boundary/load misinterpretation
No-TC	40.0	24.6-57.7	Boundary/load misinterpretation

FP: Full prompt (full system instruction); No-CoT: removal of chain-of-thought guidance; No-FS: removal of few-shot examples; No-TC: removal of tool-interface documentation.

thermal conduction. This result is important because the agent is not directly optimizing the structure; instead, it must translate user intent into a complete set of mesh, boundary-condition, material, constraint, and solver inputs. For incomplete descriptions, the agent can use geometry information and its structured instruction to infer unspecified modeling details. The agreement with classical benchmark behavior therefore provides evidence that the formulation layer, in addition to the underlying numerical solver, is functioning as intended.

The extended cases clarify where the framework adds practical value beyond regular domains. Stress-constrained examples test whether additional physical constraints can be introduced through the same tool interface, while the arbitrary-geometry cases test whether supports, loads, and passive domains can be assigned on engineering meshes. The image-to-optimization example further shows that a geometry and mesh reconstructed from a 2D image input can be connected to the same TO pipeline. These cases indicate that TO-Master is most useful when different geometry sources and problem types can be reduced to verified solver inputs.

The ablation results show that the main failure mode under fuzzy prompts is problem formulation rather than numerical optimization. Since all instruction variants use the same solver backend, the reduced success rates of No-FS, No-CoT, and No-TC mainly reflect errors in interpreting spatial descriptions, selecting valid tool arguments, or preserving required supports, loads, passive domains, and optimization parameters. The full instruction achieved the highest observed success rate, and the recorded failures illustrate different types of formulation error across the ablation configurations.

The boundary-condition visualization and user confirmation step are therefore not only usability features, but part of the reliability mechanism. Intermediate checking helps catch errors before optimization is launched and keeps the LLM agent from silently committing to an incorrect model. Assumptions introduced during inference should be made explicit for user confirmation. Conflicting requirements call for clarification of design priorities. Experts can help resolve ambiguities in supports, loads, or material models and guide parameter adjustments when the optimization results do not meet design expectations.

Several limitations remain. The ablation study is limited to six regular benchmark cases, where the problem specifications can be systematically varied and repeated. The stress-constrained, arbitrary-geometry, and image-to-optimization examples involve more case-specific geometry processing and boundary-condition definition; in these cases, users may need to inspect the previewed supports, loads, and passive domains and then confirm or adjust them before optimization. Therefore, these examples are used as capability demonstrations rather than simple repeated ablation benchmarks. The current system does not automatically convert units or check dimensional consistency; users must provide inputs in a consistent unit system. Some ablation outcomes still require manual inspection.

Future extensions could use LLMs to interpret simulation feedback and explore alternative formulations. Nonconvergence, constraint violations, and stress concentrations could guide proposed revisions, with changes recorded and confirmed before another analysis. Further directions include comparing design trade-offs, incorporating manufacturing constraints, and assessing robustness to uncertain loads and material properties.

CONCLUSIONS

This work presents TO-Master, a tool-orchestrated LLM-agent system for automated topology optimization. Across the benchmark and engineering examples, TO-Master constructs solver-ready TO problems from natural language specifications and optional mesh, geometry, or image inputs. The examples cover regular

2D and 3D structural compliance minimization, thermal conduction, multi-load structural compliance minimization, stress-constrained optimization, arbitrary engineering geometries, and a 2D-image-to-3D-geometry case, showing that the same agent-tool interface can support both standard benchmarks and more practical modeling settings.

TO-Master's contribution lies in combining automated model construction with explicit mesh and boundary-condition verification across diverse geometry inputs and TO formulations. The agent handles problem formulation and workflow orchestration, while predefined numerical tools perform the finite element analyses, sensitivity calculations, and optimization.

The ablation study compares formulation success under different instruction configurations on six benchmark cases with fuzzy natural language prompts. The full instruction achieves the highest observed success rate, while removing tool-interface documentation produces the largest decrease. The difference between FP and No-CoT is not statistically significant. The recorded failures primarily involve boundary/load interpretation, passive-domain assignment, and tool-argument construction.

Future work will further extend TO-Master from controlled benchmark settings to more open-ended engineering optimization tasks. One direction is to support richer human-agent interaction during geometry preparation, boundary-condition definition, and result assessment, so that users can iteratively refine supports, loads, passive domains, and design requirements based on previewed models and reported convergence histories. Another direction is to make the tool layer more extensible, allowing new objectives, constraints, physics modules, and manufacturing requirements to be incorporated with less problem-specific implementation effort. Finally, broader industrial case studies and more standardized evaluation protocols are needed to assess not only the optimized result, but also the correctness, robustness, and reproducibility of the full LLM-agent-driven TO formulation process.

An earlier version of this work was posted as a preprint[44].

DECLARATIONS

Authors' contributions

Conceptualization: Lin, H.; Xue, T.
Methodology: Lin, H.; Li, X.; Xu, T.
Software: Lin, H.; Zhang, W.; Xu, W.
Validation: Lin, H.; Zhang, W.; Li, X.; Xu, T.
Formal analysis: Lin, H.; Zhang, W.; Xu, T.
Investigation: Lin, H.; Xu, T.
Writing - original draft: Lin, H.; Xu, T.
Writing - review and editing: Lin, H.; Li, X.; Xu, T.; Xue, T.
Visualization: Lin, H.; Zhang, W.
Supervision: Xu, T.; Xue, T.
Funding acquisition: Xue, T.

Availability of data and materials

The TO-Master source code and numerical data supporting the reported results are retained by the authors and are available upon reasonable request from the corresponding authors. The system instructions, case specifications, and numerical settings are provided in the manuscript and the Supplementary Materials. The TO-Master application is available at https://www.bohrium.com/en/apps/to-master.

AI and AI-assisted tools statement

During manuscript preparation, DeepSeek-V4-Flash (released 2026-04-24) was used as the agent component of TO-Master for the original evaluations, and DeepSeek-V4.1-Flash-1017 (released 2026-10-17) was used for

the supplementary repeatability experiments. TripoSR (code revision d26e331, dated 2024-05-14) was used for image-to-3D surface reconstruction during the development and evaluation reported in this manuscript. The geometry reconstructed using TripoSR was used for Case 11, whose boundary conditions and optimized result are shown in Figure 6E and Figure 7E, respectively. In addition, OpenAI Codex was used to polish author-prepared text, and GPT-5.5 (released 2026-04-23) was used through OpenAI Codex to generate Scalable Vector Graphics (SVG) code for Figures 1-3. These figures were subsequently edited and finalized by the authors in Figma. This manuscript-preparation assistance did not influence the study design, data collection, analysis, interpretation, or the scientific content of the work. All authors take full responsibility for the accuracy, integrity, and final content of the manuscript.

Financial support and sponsorship

We acknowledge the support from the Research Grants Council of Hong Kong through the project (Ref No. 26205024), the Hong Kong Innovation and Technology Support Program (ITS/032/23), the HKUST Bridge Gap Fund (Ref. code: BGF.2026.019), and the research donation from Beijing DP Technology Co., Ltd.

Conflicts of interest

Xue, T. is an Editorial Board Member of the journal *Journal of Mechanoinformatics* and was not involved in any steps of editorial processing, notably including reviewers' selection, manuscript handling and decision making. Li, X. is affiliated with China Iron and Steel Research Institute Group, while the other authors have declared that they have no conflicts of interest.

Consent for publication

Not applicable.

Ethical approval and consent to participate

Not applicable.

Copyright

© The Author(s) 2026.

Supplementary Materials

Supplementary Materials

REFERENCES

1. Bendsøe, M. P.; Kikuchi, N. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods. Appl. Mech. Eng.* **1988**, *71*, 197-224. DOI
2. Zhou, M.; Rozvany, G. The COC algorithm, Part II: topological, geometrical and generalized shape optimization. *Comput. Methods. Appl. Mech. Eng.* **1991**, *89*, 309-36. DOI
3. Rozvany, G. I. N.; Zhou, M.; Birker, T. Generalized shape optimization without homogenization. *Struct. Optim.* **1992**, *4*, 250-2. DOI
4. Bendsøe, M. P.; Sigmund, O. Material interpolation schemes in topology optimization. *Arch. Appl. Mech.* **1999**, *69*, 635-54. DOI
5. Wang, M. Y.; Wang, X.; Guo, D. A level set method for structural topology optimization. *Comput. Methods. Appl. Mech. Eng.* **2003**, *192*, 227-46. DOI
6. Allaire, G.; Jouve, F.; Toader, A. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.* **2004**, *194*, 363-93. DOI
7. Xie, Y.; Steven, G. A simple evolutionary procedure for structural optimization. *Comput. Struct.* **1993**, *49*, 885-96. DOI
8. Yang, X. Y.; Xie, Y. M.; Steven, G. P.; Querin, O. M. Bidirectional evolutionary method for stiffness optimization. *AIAA. J.* **1999**, *37*, 1483-8. DOI
9. Chandrasekhar, A.; Sridhara, S.; Suresh, K. AuTO: a framework for automatic differentiation in topology optimization. *Struct. Multidiscip. Optim.* **2021**, *64*, 4355-65. DOI
10. Xue, T.; Liao, S. Gan, Z., et al. JAX-FEM: a differentiable GPU-accelerated 3D finite element solver for automatic inverse design and mechanistic data science. *Comput. Phys. Commun.* **2023**, *291*, 108802. DOI

11. Wei, J.; Wang, X. Schuurmans, D., et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35*, New Orleans, LA, USA, November 28-December 9, 2022; Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; Oh, A., Eds.; Neural Information Processing Systems Foundation, Inc. (NeurIPS): San Diego, CA, USA, 2022; pp 24824-37. DOI
12. Yao, S.; Zhao, J. Yu, D., et al. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations*, Kigali, Rwanda, May 1-5, 2023. Available online: https://openreview.net/forum?id=WE_vluYUL-X. (accessed 24 November 2026).
13. Schick, T.; Dwivedi-Yu, J. Dessi, R., et al. Toolformer: language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA, USA, December 10-16, 2023; Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; Levine, S., Eds.; Neural Information Processing Systems Foundation, Inc. (NeurIPS): San Diego, CA, USA, 2023; pp 68539-51. DOI
14. Guo, J.; Park, C.; Qian, D.; Hughes, T. J.; Liu, W. K. Large language model-empowered next-generation computer-aided engineering. *Comput. Methods. Appl. Mech. Eng.* **2026**, *450*, 118591. DOI
15. Liang, H.; Talebi Kalaleh, M.; Mei, Q. Integrating large language models for automated structural analysis. *arXiv* **2025**, arXiv:2504.09754. Available online: https://arxiv.org/abs/2504.09754. (accessed 24 November 2026).
16. Fan, E.; Hu, K. Wu, Z., et al. ChatCFD: a large language model-driven agent for end-to-end computational fluid dynamics automation with structured knowledge and reasoning. *Adv. Intell. Discov.* **2026**, *2*, e202500174. DOI
17. Pandey, S.; Xu, R.; Wang, W.; Chu, X. OpenFOAMGPT: a retrieval-augmented large language model (LLM) agent for OpenFOAM-based computational fluid dynamics. *Phys. Fluids.* **2025**, *37*, 035120. DOI
18. Zhang, L.; Le, B.; Akhtar, N.; Lam, S.; Ngo, D. Large language models for computer-aided design: a survey. *ACM. Comput. Surv.* **2026**, *58*, 1-39. DOI
19. Schüpbach, A.; San Miguel, R.; Ferchow, J.; Meboldt, M. From text to design: a framework to leverage LLM agents for automated CAD generation. *Proc. Des. Soc.* **2025**, *5*, 1893-902. DOI
20. Ha, D. Q.; Carstensen, J. V. Human-informed topology optimization: interactive application of feature size controls. *Struct. Multidiscip. Optim.* **2023**, *66*, 59. DOI
21. Schiffer, G.; Ha, D. Q.; Carstensen, J. V. HiTop 2.0: combining topology optimisation with multiple feature size controls and human preferences. *Virtual. Phys. Prototyp.* **2023**, *18*, e2268603. DOI
22. Schiffer, G.; Schmidt, M.; Pedersen, C. B. W.; Carstensen, J. V. Interactive infill topology optimisation guided by user-drawn patterns. *Virtual. Phys. Prototyp.* **2024**, *19*, e2361864. DOI
23. Ha, D. Q.; Alam, M. F.; Buehler, M. J.; Ahmed, F.; Carstensen, J. V. AI-guided human-in-the-loop inverse design of high performance engineering structures. *Commun. Eng.* **2026**, Epub ahead of print. DOI
24. Yang, S.; Wang, J.; Wang, Y. Large language models as optimization controllers: adaptive continuation for SIMP topology optimization. *Adv. Eng. Softw.* **2026**, *Epub ahead of print*. DOI
25. Yang, S.; Wang, J.; Wang, Y. AutoSiMP: autonomous topology optimization from natural language via LLM-driven problem configuration and adaptive solver control. *arXiv* **2026**, arXiv:2603.27000. Available online: https://arxiv.org/abs/2603.27000. (accessed 24 November 2026).
26. Stewart, I. A.; Chen, H.; Ahmed, F. TO-Agents: a multi-agent AI pipeline for preference-guided topology optimization. *arXiv* **2026**, arXiv:2605.21622. Available online: https://arxiv.org/abs/2605.21622. (accessed 24 November 2026).
27. Liang, Z.; Zhang, Y.; Wang, Y.; Li, W. Integrating large models with topology optimization for conceptual design realization. *Adv. Eng. Inform.* **2025**, *67*, 103524. DOI
28. Park, H.; Chung, H. Self-refining topology optimization via an LLM-based multi-agent framework. *arXiv* **2026**, arXiv:2605.23273. Available online: https://arxiv.org/abs/2605.23273. (accessed 24 November 2026).
29. Bendsøe, M. P.; Sigmund, O. *Topology Optimization*; Springer Berlin Heidelberg, 2004. DOI
30. Bourdin, B. Filters in topology optimization. *Int. J. Numer. Methods. Eng.* **2001**, *50*, 2143-58. DOI
31. Guest, J. K.; Prévost, J. H.; Belytschko, T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int. J. Numer. Methods. Eng.* **2004**, *61*, 238-254. DOI
32. Wang, F.; Lazarov, B. S.; Sigmund, O. On projection methods, convergence and robust formulations in topology optimization. *Struct. Multidiscip. Optim.* **2011**, *43*, 767-84. DOI
33. Yang, D.; Liu, H.; Zhang, W.; Li, S. Stress-constrained topology optimization based on maximum stress measures. *Comput. Struct.* **2018**, *198*, 23-39. DOI
34. Le, C.; Norato, J.; Bruns, T.; Ha, C.; Tortorelli, D. Stress-based topology optimization for continua. *Struct. Multidiscip. Optim.* **2010**, *41*, 605-20. DOI

35. Gersborg-Hansen, A.; Bendsøe, M. P.; Sigmund, O. Topology optimization of heat conduction problems using the finite volume method. *Struct. Multidiscip. Optim.* **2006**, *31*, 251-9. DOI
36. Svanberg, K. The method of moving asymptotes—a new method for structural optimization. *Int. J. Numer. Methods. Eng.* **1987**, *24*, 359-73. DOI
37. Zhang, L.; Chen, S. Cai, Y., et al. Bohrium + SciMaster: building the infrastructure and ecosystem for agentic science at scale. *AI. Plus.* **2026**, *1*, 100005. DOI
38. Google. *Agent Development Kit (ADK)*. Available online: https://google.github.io/adk-docs/. (accessed 24 November 2026).
39. Model context protocol contributors. *Model Context Protocol documentation*. Available online: https://modelcontextprotocol.io/docs/getting-started/intro. (accessed 24 November 2026).
40. Geuzaine, C.; Remacle, J. F. Gmsh: a 3D finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods. Eng.* **2009**, *79*, 1309-31. DOI
41. Tochilkin, D.; Pankratz, D. Liu, Z., et al. TripoSR: fast 3D object reconstruction from a single image. *arXiv* **2024**, arXiv:2403.02151. Available online: https://arxiv.org/abs/2403.02151. (accessed 24 November 2026).
42. GrabCAD. *GE jet engine bracket challenge*. Available online: https://grabcad.com/challenges/ge-jet-engine-bracket-challenge. (accessed 24 November 2026).
43. GrabCAD. *Airplane bearing bracket challenge*. Available online: https://grabcad.com/challenges/airplane-bearing-bracket-challenge. (accessed 24 November 2026).
44. Lin, H.; Zhang, W.; Xu, W.; Li, X.; Xu, T.; Xue, T. TO-Master: an LLM-agent framework for automated topology optimization. *arXiv* **2026**, arXiv:2607.01812. Available online: https://arxiv.org/abs/2607.01812. (accessed 24 November 2026).

Disclaimer/Publisher's Note: All statements, opinions, and data contained in this publication are solely those of the individual author(s) and contributor(s) and do not necessarily reflect those of OAE and/or the editor(s). OAE and/or the editor(s) disclaim any responsibility for harm to persons or property resulting from the use of any ideas, methods, instructions, or products mentioned in the content.



© The Author(s) 2026. Open Access This article is licensed under a Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.