TRUSSORBIT: EDGE-ORBIT SEQUENCE MODELING FOR TRUSS METAMATERIAL INVERSE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Inverse design of periodic truss metamaterials requires accurate stiffness matching, physically valid generation, and a representation that spans diverse topologies. We present TrussOrbit, a structural language model conditioned on nine effective-stiffness components. Its Edge Orbit Truss Representation (EOTR) encodes node geometry and connectivity through Pmmm symmetry orbits. A unified data pipeline consolidates four public sources into 1.49 million canonical structures with common finite-element labels. We introduce Setwise Marginal Credit Optimization (SMCO) to optimize stiffness matching and binary validity across fixed-size candidate groups through sequence-level marginal credit. On the unified benchmark, TrussOrbit achieves the lowest NMAE, NRMSE, and MAPE among the compared models while covering the full dataset; it generates candidates 6.6 times as fast as the fastest external baseline. Representation and policy-objective ablations show complementary gains: EOTR shortens sequences and reduces invalid generation, while SMCO improves stiffness matching over supervised fine-tuning in both single-candidate and best-of-16 evaluation.

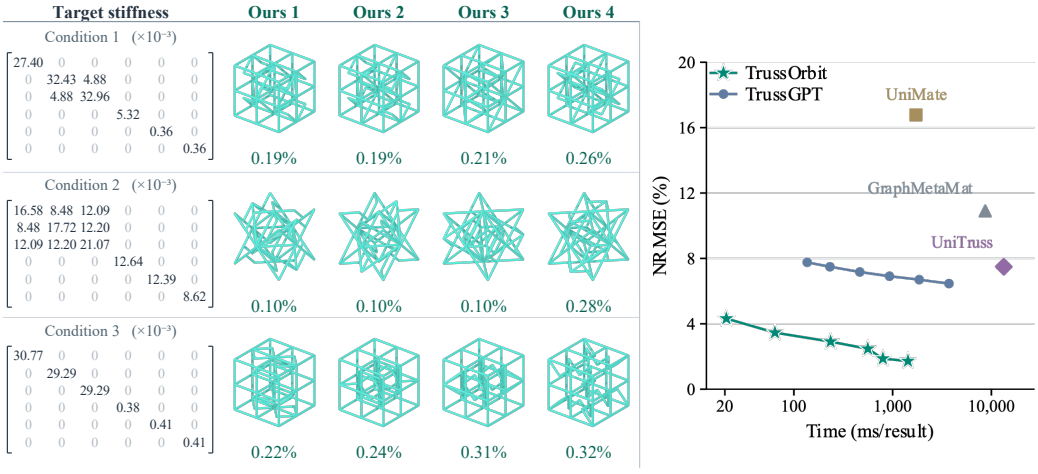


Figure 1: **Left:** Four generated structures for each of three target stiffness matrices; labels below the structures report NRMSE. **Right:** Best-of-k NRMSE versus generation time for $k = 1, 2, 4, 8, 16, 32$; stars denote TrussOrbit. Time reports all k candidates per target and excludes model loading and FEM evaluation.

1 INTRODUCTION

Mechanical metamaterials obtain programmable macroscopic properties through microstructural design. Periodic truss networks offer a design space for lightweight load-bearing structures, energy absorption, vibration control, and biomechanical systems (Bastek et al., 2022; Zheng et al., 2023a; Maurizi et al., 2025). Truss inverse design maps a target mechanical response to a periodic unit

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

cell, providing a promising route from desired functionality to realizable material architectures. Nevertheless, this is a non-trivial inverse problem.

Existing approaches broadly fall into two categories: physics-based optimization and learning-based generation. Physics-based methods repeatedly simulate candidates for each target, making inverse design computationally expensive (Dos Reis & Karathanasopoulos, 2022; Maurizi et al., 2022). Learning-based methods use direct generation or latent-space search (Bastek et al., 2022; Zheng et al., 2023a; Maurizi et al., 2025; Zhan et al., 2025). Recently, autoregressive methods such as TrussGPT (Lin et al., 2026) have represented truss structures as sequences and generated them token by token, providing a unified framework for modeling node geometry and connectivity while enabling direct conditional generation. Nevertheless, two limitations remain: (1) listing every node and strut in a full unit cell repeats symmetry-related information, lengthens decoding, and increases exposure to cascading token errors that can yield invalid structures; and (2) paired supervision overlooks the one-to-many nature of inverse design, in which multiple structures can satisfy the same stiffness target.

To overcome these limitations, our core idea is to represent a periodic truss through symmetry orbits: the model generates representative node coordinates and edge connections, and symmetry operations recover the full cell. This shifts sequence modeling from full-cell enumeration to orbit-level geometry and connectivity, while setwise physical feedback supervises multiple candidates for each stiffness target. However, realizing this idea presents three challenges: (1) the representation must encode variable node geometry, connectivity, and periodic symmetry in a compact, consistent, and recoverable form; (2) learning must account for the one-to-many nature of inverse design and assign meaningful credit to individual candidates according to their physical quality and validity; and (3) heterogeneous truss datasets must be reconciled across structural conventions and relabeled under a common finite-element protocol to provide a unified training corpus.

To address these challenges, we propose **TrussOrbit**, an edge-orbit approach to periodic truss inverse design. TrussOrbit uses **Edge Orbit Truss Representation (EOTR)** to encode one representative per node and edge orbit and recover complete Pmmm trusses through symmetry operations. The resulting structural sequences preserve variable geometry and connectivity while shortening autoregressive decoding. To learn from multiple candidates, **Setwise Marginal Credit Optimization (SMCO)** evaluates fixed-size groups using stiffness-matching and binary validity rewards, and assigns each sequence credit by replacing its slot with a null candidate and measuring its marginal contribution. This brings physical feedback from multiple alternatives into sequence-level policy optimization. Using EOTR as the shared interface, a unified data pipeline canonicalizes and deduplicates four public truss sources and recomputes their properties under a common finite-element protocol, producing 1,492,505 structures with consistent labels and more than 450,000 topology families. On this benchmark, single-candidate TrussOrbit reduces NRMSE by 42.3% relative to UniTruss and generates candidates $6.6\times$ as fast as TrussGPT, the fastest external baseline. Figure 1 shows its speed–accuracy advantage and multiple structures matching each target stiffness.

Our contributions are:

1. We propose **TrussOrbit**, which represents node geometry and strut connectivity through **EOTR**. Encoding symmetry orbits yields compact sequences that recover complete Pmmm trusses, reducing decoding cost and invalid generation.
2. We propose **SMCO**, which assigns sequence-level marginal credit within fixed-size candidate groups using physical property matching and validity, improving both single-candidate and best-of-16 performance over supervised fine-tuning.

2 Related Work

2.1 Inverse design of truss metamaterials

Truss inverse design controls topology and geometry through homogenization, neural surrogates, and generative search (Zheng et al., 2021; Dos Reis & Karathanasopoulos, 2022; Deng et al., 2022; Maurizi et al., 2022). Parameterizations include implicit fields, Voronoi cells, and explicit struts (Wang et al., 2022; Zheng et al., 2023b; Dos Reis & Karathanasopoulos, 2024). Zheng et al. (2026) tokenize basis functions, coefficients, and operators of implicit equations for periodic shell

isosurfaces. Their trigonometric basis defines periodic surfaces without node and strut records; EOTR instead stores variable node-orbit coordinates and edge connections with explicit relative Pmmm operations. Topology posteriors and graph latent models support variable connectivity (Bastek et al., 2022; Zheng et al., 2023a; Kim et al., 2024), and TrussGPT serializes trusses for language-model generation (Lin et al., 2026). Related studies address nonlinear response, piezoelectricity, and multiple materials (Ha et al., 2023; Sharma et al., 2025; Frey et al., 2025). EOTR pairs a common Pmmm orbit sequence with unified FEM labels.

2.2 Generative models for inverse design

Generative inverse design models a distribution of structures conditioned on a desired response. Conditional latent models and normalizing flows provide sampleable design spaces (Ma et al., 2019; Yang et al., 2023); conditional VAEs target bandgap properties (Tran et al., 2025), while UniMate uses diffusion for mechanical design (Zhan et al., 2025). GraphMetaMat combines autoregressive graph generation with physical feedback (Maurizi et al., 2025), and graph-latent diffusion supports curved cellular structures (Abu-Mualla & Huang, 2025). Structural sequences and physics-guided generation further connect sampled designs to their responses (Li et al., 2026b; Wang et al., 2026; Xue et al., 2025). These approaches combine a structural parameterization with a mechanism for incorporating numerical property targets. TrussOrbit conditions discrete orbit sequences on a continuous stiffness prefix, keeping the property input separate from the structural vocabulary and sharing one decoder across the imported Pmmm datasets.

2.3 Multiplicity in inverse design

Multiple structures can realize similar responses. Invertible networks, diversity-regularized generation, and diffusion posterior methods address this ambiguity through latent variables and alternative candidates (Ardizzone et al., 2019; Heyrani Nobari et al., 2021; Chung et al., 2023; Cardoso et al., 2024), including in mechanical inverse design (Bastek et al., 2022; Bastek & Kochmann, 2023; Nakarmi et al., 2026; Zheng et al., 2026). Photonic inverse design likewise obtains alternative solutions through mixture-density models and conditional generation (Unni et al., 2020; Yeung et al., 2021; Dai et al., 2022). This multiplicity motivates evaluating both target-response accuracy and the availability of alternative structures. Multi-sample policy objectives connect training to inference budgets (Tang et al., 2025; Walder & Karkhanis, 2025), and SetPO assigns marginal credit for set diversity (Li et al., 2026a). SMCO applies marginal credit to physical matching and validity through null-slot comparisons that retain the candidate budget. EOTR supplies the representation of alternative structures, while SMCO optimizes their contribution to a sampled group.

3 Preliminaries

3.1 Truss representation

A periodic truss unit cell is represented by $\mathcal{T} = (V, \mathcal{E}, X)$, where $V = \{v_i\}_{i=1}^{|V|}$ is its node set, $\mathcal{E}$ is a set of undirected node pairs defining struts, and $X \in [0,1]^{|V|\times 3}$ contains normalized node coordinates. In the Pmmm domain, coordinate reflections relate symmetry-equivalent nodes and struts. Applying the symmetry operations to one representative generates its orbit, with coincident images merged. Periodic assembly identifies lattice-equivalent entities on the cell boundary.

Under linear elasticity, the effective stiffness of a Pmmm truss has nine independent components in Voigt notation (Zheng et al., 2023a). We collect them in

$$\mathbf{c}(\mathcal{T}) = (C_{11}, C_{22}, C_{33}, C_{12}, C_{13}, C_{23}, C_{44}, C_{55}, C_{66}) \in \mathbb{R}^9. \tag{1}$$

Here C_{ab} denotes entry (a, b) of the effective stiffness matrix, and $\mathbf{c}^\star$ denotes an input target. This component order is fixed throughout conditioning and evaluation. Appendix C gives the full matrix, periodic weighting, radius calculation, and stiffness extraction.

3.2 Autoregressive modeling

Let $\mathbf{s} = (s_1, \ldots, s_L)$ be a token sequence with length L, including its end marker, and let $s_t \in \mathcal{V}$ belong to a discrete vocabulary $\mathcal{V}$. For condition $\mathbf{c}$, an autoregressive model with parameters θ

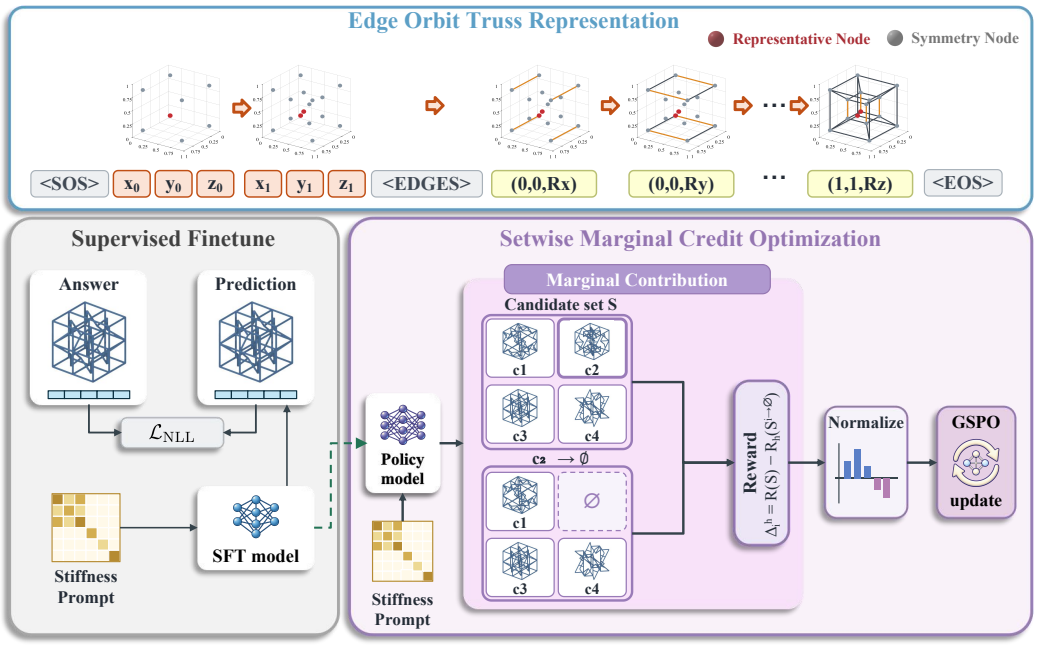


Figure 2: Overview of TrussOrbit. Edge Orbit Truss Representation (EOTR) encodes representative nodes and edge orbits, then reconstructs the full truss by symmetry expansion. Supervised fine-tuning (SFT) learns stiffness-conditioned sequences using $\mathcal{L}_{\text{NLL}}$. Setwise Marginal Credit Optimization (SMCO) samples candidate groups from the SFT model. Marginal contribution measures the change in stiffness-matching and validity rewards when one candidate becomes a null slot ($\varnothing$). Group-normalized credits drive the Group Sequence Policy Optimization (GSPO) update.

factorizes the sequence probability as

$$p_\theta(\mathbf{s} \mid \mathbf{c}) = \prod_{t=1}^{L} p_\theta(s_t \mid \mathbf{s}_{<t}, \mathbf{c}), \qquad \mathbf{s}_{<t} = (s_1, \ldots, s_{t-1}). \tag{2}$$

For sample i, the sequence and its length are denoted by $\mathbf{s}_i$ and L_i. Likelihood training predicts each token from its observed prefix, whereas inference extends a sampled prefix. Omitting $\mathbf{c}$ gives the unconditional model. This autoregressive formulation underlies decoder-only language models and structural sequence generation (Lin et al., 2026); the next section specifies the truss tokens and stiffness interface.

4 Method

Throughout this paper, **TrussOrbit** (**Ours**) denotes the complete EOTR model after structure pre-training, SFT, and SMCO; ablation tables label it **SFT+SMCO**. **Edge Orbit Truss Representation (EOTR)** encodes periodic trusses as sequences of node and edge orbits; a stiffness conditioner maps the target stiffness vector to continuous prefix embeddings for autoregressive generation, and symmetry expansion reconstructs the full structure. **Setwise Marginal Credit Optimization (SMCO)** uses FEM feedback to assign sequence-level credit within fixed-size candidate groups according to stiffness matching and physical validity. The TrussOrbit dataset unifies four public truss sources through EOTR canonicalization, deduplication, and stiffness relabeling under a common FEM protocol, yielding 1,492,505 labeled structures. Figure 2 illustrates EOTR and the SFT and SMCO stages.

4.1 PROBLEM FORMULATION

Let $\mathcal{S}_{\mathrm{Pmmm}}$ denote the set of periodically connected Pmmm trusses whose finite-element method (FEM) stiffness evaluation succeeds under the fixed material and density settings of Section 5.1. Let d be a nonnegative discrepancy function on pairs of nine-component stiffness vectors. The inverse-design objective is

$$\mathcal{T}^\star \in \operatorname*{arg\,min}_{\mathcal{T}\in\mathcal{S}_{\mathrm{Pmmm}}} d(\mathbf{c}(\mathcal{T}), \mathbf{c}^\star)\,. \tag{3}$$

Here $\mathcal{T}^\star$ denotes any minimizer in $\mathcal{S}_{\mathrm{Pmmm}}$, and $\arg\min$ denotes the set of minimizers. Multiple geometries and topologies can yield similar target stiffness (Bastek et al., 2022; Zheng et al., 2023a). We therefore learn a conditional sequence distribution, sample candidate structures, and evaluate their stiffness and validity with FEM; returned candidates are assessed by their achieved discrepancy.

4.2 EDGE ORBIT TRUSS REPRESENTATION

Serializing every node and strut repeats symmetry-related information and lengthens generation. **Edge Orbit Truss Representation (EOTR)** stores one representative per node and strut orbit, then reconstructs the full Pmmm cell through symmetry operations. Shared canonical keys make the encoding independent of input node labels and edge-list order.

Let $\bar{V} = \{\bar{v}_i\}_{i=1}^{N_v}$ be the ordered node-orbit representatives and let N_e be the number of edge orbits. The serialized representation is

$$\mathrm{EOTR}(\mathcal{T}) = [\langle\mathrm{SOS}\rangle, \bar{v}_1, \ldots, \bar{v}_{N_v}, \langle\mathrm{EDGES}\rangle, (U_{i_m}, \mathrm{VGH}_{j_m,h_m})_{m=1}^{N_e}, \langle\mathrm{EOS}\rangle]. \tag{4}$$

The special token $\langle\mathrm{SOS}\rangle$ (start of sequence) marks the beginning of the truss sequence; $\langle\mathrm{EDGES}\rangle$ separates the node-coordinate block from the edge-orbit records and marks the start of connectivity encoding; and $\langle\mathrm{EOS}\rangle$ (end of sequence) marks completion of the truss sequence and terminates generation. Each $\bar{v}_i$ is represented by three coordinate tokens in (x, y, z) order, using the shared 1,025-point grid $\{j/1024 : j = 0, \ldots, 1024\}$.

Let g_h, $h = 0, \ldots, 7$, denote the eight Pmmm point-group operations after quotienting out periodic translation. In edge-orbit record m, U_{i_m} encodes the source node-orbit index, while VGH_{j_m,h_m} jointly encodes the target index and relative operation. The two tokens specify a strut from $\bar{v}_{i_m}$ to $g_{h_m}(\bar{v}_{j_m})$; applying all operations recovers its edge orbit. Thus, each coordinate contributes three tokens and each edge orbit contributes two, independent of the number of symmetry-related struts reconstructed in the full cell. Canonical ordering makes equivalent input node labels and edge-list permutations map to the same sequence. The stored sequence length is therefore $L_{\mathrm{EOTR}} = 3N_v + 2N_e + 3$, including SOS, EDGES, and EOS. Appendix A details canonicalization and the codec.

4.3 SETWISE MARGINAL CREDIT OPTIMIZATION

Likelihood training supervises individual structures, whereas inverse design benefits from several candidates that match the same target. Inspired by SetPO's marginal set-credit principle (Li et al., 2026a), **SMCO** evaluates physical quality across a candidate group and assigns each sequence credit for its contribution. For target $\mathbf{c}^\star$, the frozen old policy $p_{\theta_{\mathrm{old}}}$ independently samples k sequences, giving $\mathcal{S} = ((\mathbf{s}_i, \mathcal{T}_i))_{i=1}^k$ after decoding. Repeated outputs and failures retain their slots. The reward is permutation-invariant and scores the candidate set as a whole, while null-slot counterfactuals identify each sequence's marginal contribution without changing group size. This separates group-level design quality from the per-sequence policy update.

Physical set rewards. We reward both a close stiffness match and a high fraction of usable candidates. Let $v_i^{\mathrm{val}} \in \{0, 1\}$ indicate successful parsing, structural checks, and FEM evaluation. For valid candidates, define $q_i = \exp(-\,\mathrm{NRMSE}_i^{\mathrm{RL}}\,/\tau_u)$ with $\tau_u > 0$; failures receive $q_i = 0$. The RL error uses component-wise training-set 5th–95th percentile widths, while evaluation uses max–min ranges (Appendix D). The set rewards are

$$R_{\mathrm{fit}}(\mathcal{S}) = \tau_b \log\left[\frac{1}{k}\sum_{i=1}^{k} e^{q_i/\tau_b}\right], \qquad R_{\mathrm{val}}(\mathcal{S}) = \frac{1}{k}\sum_{i=1}^{k} v_i^{\mathrm{val}}, \qquad \tau_b > 0. \tag{5}$$

As $\tau_b \to 0$, the fit reward approaches the group's largest quality score, linking it to multi-sample inference (Tang et al., 2025; Walder & Karkhanis, 2025). At finite temperature it smoothly shares sensitivity across high-quality candidates. The validity reward counts all requested slots, including failed decodes in its denominator, so groups receive a lower validity score when generation failures increase.

Fixed-cardinality credit. To translate group rewards into candidate credit, replace slot i by a null candidate with $q_i = v_i^{\mathrm{val}} = 0$, forming $\mathcal{S}^{0,i}$. This retains k slots and the same reward denominators, unlike deletion-based diversity credit (Li et al., 2026a). The marginal contribution is

$$\Delta_i^{(h)} = R_h(\mathcal{S}) - R_h(\mathcal{S}^{0,i}), \qquad h \in \{\mathrm{fit}, \mathrm{val}\}. \tag{6}$$

Counterfactual rewards reuse the original FEM results; no additional decoding or FEM solve is required. A candidate receives positive credit only when its presence raises the corresponding set reward relative to a null slot. Appendix B gives the exact margins, duplicate behavior, and raw on-policy credit identity.

Group-normalized advantages. To balance the two credit channels and center the learning signal, normalize each within the target group, combine them, and normalize once more. For $\mathbf{z} \in \mathbb{R}^k$, define

$$\mathcal{N}_i(\mathbf{z}) = \frac{z_i - \mathrm{mean}(\mathbf{z})}{\mathrm{std}(\mathbf{z}) + \varepsilon_A},$$

where std uses population variance and $\varepsilon_A > 0$. With validity weight $\lambda_{\mathrm{val}} \geq 0$, separate channel normalization (Liu et al., 2026) gives

$$A_i^{(h)} = \mathcal{N}_i(\boldsymbol{\Delta}^{(h)}), \qquad \widetilde{A}_i = A_i^{(\mathrm{fit})} + \lambda_{\mathrm{val}} A_i^{(\mathrm{val})}, \qquad \widehat{A}_i^{\mathrm{set}} = \mathcal{N}_i(\widetilde{\mathbf{A}}). \tag{7}$$

The vectors collect the corresponding k scalars; constant channels yield zero advantage. Separate normalization lets the validity weight control the contribution of that channel without depending on the raw units of either reward. Each $\widehat{A}_i^{\mathrm{set}}$ drives one length-normalized sequence-ratio update in Section 4.6; the update shares the resulting advantage across the sequence and clips its ratio once. Appendix B details the normalized surrogate and its relation to raw marginal credit.

4.4 Architecture

Following TrussGPT (Lin et al., 2026), we use a GPT-style decoder-only Transformer (Radford et al., 2018; Vaswani et al., 2017) to generate EOTR sequences. The decoder has 16 Transformer blocks, 16 masked self-attention heads per block, and a hidden size of 1,024; each block also contains a position-wise feed-forward network, residual connections, and normalization. Causal attention conditions each token on preceding structural tokens and, for stiffness-conditioned generation, the nine continuous prefix embeddings produced by the shared conditioner in Section 4.6. A vocabulary projection and softmax produce next-token probabilities, while the same structural-token decoder serves unconditional pretraining and conditional generation. Appendix E gives the implementation settings.

4.5 Dataset

We combine Pmmm-compatible structures from UniTruss, TrussGPT, GraphMetaMat, and PiezoTruss (Zheng et al., 2023a; Lin et al., 2026; Maurizi et al., 2025; Sharma et al., 2025); the two TrussGPT collections form five input collections across these four sources. Validity filtering leaves 3,362,319 records, and global exact-EOTR deduplication yields 1,492,505 canonical structures with nine stiffness components relabeled under the common ae108 protocol. Four-round Weisfeiler–Lehman (WL4) signatures (Shervashidze et al., 2011) group these structures into 452,791 families, each assigned wholly to one split: 1,336,666 training, 86,577 validation, and 69,262 test structures. Appendix A reports source-level filtering, deduplication, and structural coverage; Appendix C specifies the FEM settings.

4.6 Training Objective

For stiffness conditioning, we standardize each target component as $\widetilde{c}_j = (c_j - \mu_j)/\sigma_j$, $j = 1, \ldots, 9$, using the frozen training-set mean μ_j and standard deviation $\sigma_j > 0$. A component embedding layer and self-attention encoder (Vaswani et al., 2017) map the standardized vector to nine continuous prefix embeddings, prepended to the EOTR token embeddings (Lester et al., 2021). Training and inference share this conditioner; the EOTR vocabulary contains structural tokens only.

We train TrussOrbit through structure pretraining, stiffness-conditioned SFT, and SMCO. SFT minimizes next-token negative log-likelihood over N_{SFT} structure–stiffness pairs $(\mathcal{T}_i, \mathbf{c}_i)$, where $\mathbf{c}_i = \mathbf{c}(\mathcal{T}_i)$ (Lin et al., 2026):

$$\mathcal{L}_{\mathrm{NLL}}(\theta) = -\sum_{i=1}^{N_{\mathrm{SFT}}} \sum_{t=1}^{L_i} \log p_\theta(s_{i,t} \mid \mathbf{s}_{i,<t}, \mathbf{c}_i). \tag{8}$$

Here $\mathbf{s}_i = (s_{i,1}, \ldots, s_{i,L_i})$ is the scored EOTR sequence after the supplied SOS; L_i includes EOS and excludes padding and continuous prefixes. Appendix E gives the training schedules and implementation settings.

SMCO continues from the SFT model. Let $\mathcal{D}_{\mathrm{target}}$ denote the training-target distribution, and define the GSPO length-normalized sequence ratio (Zheng et al., 2025) by $\eta_i(\theta) = \exp\{L_i^{-1}[\log p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star) - \log p_{\theta_{\mathrm{old}}}(\mathbf{s}_i \mid \mathbf{c}^\star)]\}$. With clip widths $\epsilon_{\mathrm{l}}, \epsilon_{\mathrm{h}} \geq 0$, the policy objective is

$$\mathcal{J}_{\mathrm{SMCO}}(\theta) = \mathbb{E}_{\substack{\mathbf{c}^\star \sim \mathcal{D}_{\mathrm{target}} \\ \mathcal{S} \sim p_{\theta_{\mathrm{old}}}^k}} \left[\frac{1}{k} \sum_{i=1}^{k} \min\Big\{ \eta_i(\theta) \widehat{A}_i^{\mathrm{set}}, \; \mathrm{clip}(\eta_i(\theta), 1 - \epsilon_{\mathrm{l}}, 1 + \epsilon_{\mathrm{h}}) \widehat{A}_i^{\mathrm{set}} \Big\} \right]. \tag{9}$$

Here $\mathrm{clip}(z, a, b)$ truncates z to $[a, b]$, and $p_{\theta_{\mathrm{old}}}^k$ denotes k independent samples for the same target. We maximize this objective with rewards, computed advantages, and old-policy probabilities fixed during each update. Candidate interactions enter through the advantages; each sequence has one ratio and one clipping decision. Group normalization, length normalization, and clipping define the practical surrogate, whose relationship to the raw on-policy reward gradient is derived in Appendix B.

5 Experiments

5.1 Experimental setup

Implementation details. We evaluate all methods with ae108 using circular Timoshenko beams at $E = 1$, $\nu = 0.3$, and relative density $\rho = 0.15$ (Zheng et al., 2023a; Mechanics and Materials Lab, 2020). TrussOrbit uses structure pretraining, stiffness-conditioned SFT, and SMCO with 32 candidates per training target. Appendices C and E detail the solver, training schedules, and implementation settings.

Metrics. On FEM-successful responses, NMAE and NRMSE use frozen training-set component-wise max–min ranges; MAPE masks exact-zero target components. Dataset coverage is the fraction of 1,492,505 canonical structures admitted by a method's native truss representation and adapter under its vertex-coordinate and connectivity constraints. In Table 1, Validity is valid candidates divided by all requests for single-candidate rows; for best-of-16, it is targets with at least one valid candidate divided by all requested targets. Its 100% means each target yielded a valid candidate within 16 attempts. Time excludes model loading and FEM; Appendix D gives the full definitions.

Baselines. UniMate, UniTruss, GraphMetaMat, and TrussGPT cover diffusion, latent optimization, graph generation, and structural-language approaches (Zhan et al., 2025; Zheng et al., 2023a; Maurizi et al., 2025; Lin et al., 2026). We retain their native generation protocols and evaluate outputs with the common FEM pipeline; Table 1 reports their different dataset coverage. A separate policy comparison uses SFT, RLOO, GRPO, GSPO, SetPO, and SMCO (Ahmadian et al., 2024; Shao et al., 2024; Zheng et al., 2025; Li et al., 2026a). The baseline objectives and implementation settings appear in Appendices B and E.

Table 1: Inverse-design comparison under the unified FEM protocol. Dataset coverage measures native representation and adapter support over the canonical truss corpus.

Method	Validity (%) ↑	Time (ms/result) ↓	Dataset coverage (%) ↑	NMAE ↓	NRMSE ↓	MAPE ↓
UniMate (Zhan et al., 2025)	**100.00**	1,720.99	2.93	0.1403	0.1678	1.0000
UniTruss (Zheng et al., 2023a)	37.70	13,376.47	64.11	0.0597	0.0749	0.4159
GraphMetaMat (Maurizi et al., 2025)	75.50	8,668.08	9.23	0.0889	0.1090	0.6836
TrussGPT (Lin et al., 2026)	1.18	136.48	35.90	0.0609	0.0776	0.3073
TrussOrbit (Ours)	93.27	**20.70**	**100.00**	0.0354	0.0432	0.2623
TrussOrbit (best-of-16)	**100.00**	796.71	**100.00**	**0.0133**	**0.0187**	**0.1058**

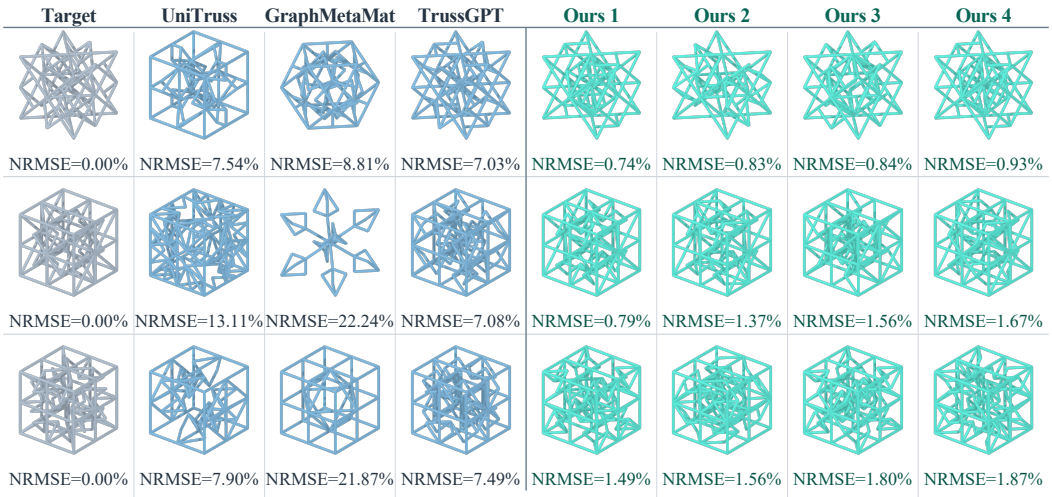


Figure 3: Qualitative inverse-design comparison. Rows from top to bottom correspond to different target stiffnesses. Ours 1–4 show four TrussOrbit candidates per target; labels report NRMSE.

5.2 Comparison

5.2.1 Quantitative analysis

Table 1 compares stiffness errors, generation cost, and dataset coverage. Property errors use FEM-successful responses; best-of-16 selects the valid candidate with the lowest NRMSE for each target.

Single-candidate TrussOrbit obtains 0.0432 NRMSE with 93.27% validity, outperforming the external baselines on NRMSE. It also achieves the best NMAE and MAPE among the compared models. UniTruss is the strongest external comparator for NMAE, while TrussGPT is strongest on MAPE.

TrussOrbit's 100% dataset coverage spans all 452,791 WL4 topology families, whereas the external representations admit 2.93–64.11% of canonical structures. TrussOrbit also generates candidates 6.59× as fast as the fastest external baseline under native procedures; timing excludes model loading and FEM evaluation.

5.2.2 Qualitative analysis

Figure 3 shows displayed TrussOrbit candidates with NRMSE below 2%. They retain close stiffness matches while exhibiting different internal connectivity; the displayed baseline outputs have substantially larger errors. These selected cases illustrate multiple structural solutions for the same target alongside the aggregate results in Table 1.

Table 2: Ablations on EOTR representation (top) and policy objective (bottom). Length is tokens per structure; time is milliseconds per structure; validity is the valid-generation fraction.

(a) EOTR representation

Encoding	Length (tokens) ↓	Time (ms) ↓	Validity (%) ↑
Node pair	740.79	96.37	88.15
Node index pair	450.19	52.46	90.95
Edge orbit	**75.07**	**23.28**	**95.39**

(b) Policy objective

Method	Validity (%) ↑	NMAE ↓	NRMSE ↓	MAPE ↓
SFT	92.64	0.0371	0.0479	0.2736
SFT+RLOO	92.39	0.0410	0.0526	0.2717
SFT+GRPO	92.46	0.0409	0.0569	0.2672
SFT+GSPO	92.72	0.0361	0.0544	0.2663
SFT+SetPO	92.87	0.0387	0.0522	0.2670
SFT+SMCO (Ours)	**93.27**	**0.0354**	**0.0432**	**0.2623**

5.3 Ablation study

Representation ablation. In the top panel of Table 2, the separately trained encodings share a backbone, training split, and decoding configuration. Relative to node-index pairs, edge-orbit encoding reduces mean sequence length by 83.3% and generation time by 55.6%, raises validity from 90.95% to 95.39%, and lowers the invalid-generation fraction by 49.1%.

Policy-objective ablation. The bottom panel compares the SFT checkpoint with its RLOO, GRPO, GSPO, SetPO, and SMCO continuations (Ahmadian et al., 2024; Shao et al., 2024; Zheng et al., 2025; Li et al., 2026a) on the same targets with a common single-candidate test-time budget, decoding configuration, validity checks, and FEM evaluator. Training group sizes and optimizer settings follow the respective official implementations (Appendix E). TrussOrbit has the highest validity and lowest NMAE, NRMSE, and MAPE, while the other objectives trade lower MAPE for higher NRMSE. At $k = 16$, TrussOrbit lowers the three errors by 13.6%, 15.0%, and 17.9% relative to SFT; both reach 100% target success, and TrussOrbit yields more distinct matched topologies (Appendix F). Across the main comparison, TrussOrbit generates candidates 6.6× as fast as TrussGPT, the fastest external baseline.

6 Conclusion

TrussOrbit combines EOTR and SMCO for stiffness-conditioned inverse design of periodic trusses, trained on 1,492,505 canonical structures with common FEM labels. It achieves 0.0432 single-candidate NRMSE and 93.27% validity, generating candidates 6.6× as fast as TrussGPT, the fastest external baseline. EOTR reduces sequence length by 83.3% and invalid generation by 49.1% relative to node-index pairs; TrussOrbit improves stiffness matching over SFT at $k = 1$ and $k = 16$ and finds more distinct matched topologies at $k = 16$.

However, high-error and invalid candidates remain (Fig. 4). The inverse-design target comprises the nine independent components of the effective stiffness matrix for Pmmm structures under fixed material properties and relative density ($E = 1$, $\nu = 0.3$, $\rho = 0.15$). Full stress–strain curves have not been evaluated as targets. Future work will consider curve-conditioned design, additional symmetry groups, variable materials and densities, and improved robustness on difficult targets.

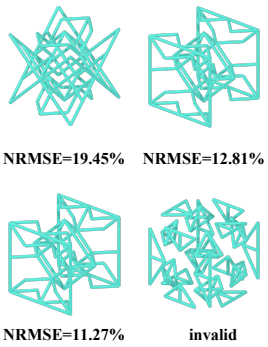


Figure 4: Three high-error structures and one invalid output.

AI USE STATEMENT

In this work, we used generative AI tools to polish writing, retrieve and identify related literature, assist with refinement of the research plan and its execution, draft portions of the manuscript, and review and refine mathematical derivations and proof arguments. The research idea, scientific narrative, and core method originated with the authors; AI assistance in ideation and execution was limited to refinement under the authors' direction. All reported numerical results come from actual experiments and were not generated by AI. The authors reviewed the AI-assisted text, references, and mathematical reasoning and take responsibility for the experimental results and the final content of the paper.

ETHICS STATEMENT

We use structural data from the cited public truss studies and evaluate generated designs using finite-element method (FEM) simulations.

REPRODUCIBILITY STATEMENT

Section 4.5 and Appendix A document dataset construction, source integration, canonicalization, and splitting. Section 4.2 and Appendix A specify the EOTR representation; Section 4.3 and Appendix B present the SMCO algorithm and its full derivation. Sections 4.4, 4.6, and 5.1, together with Appendices C–E, provide the model architecture, training settings, FEM solver, metric definitions, and evaluation protocol. Appendix F reports additional experiments and specifies the candidate records needed to reproduce them. We will release the complete code upon acceptance of this paper.

REFERENCES

Mohammad Abu-Mualla and Jida Huang. Inverse design of curved mechanical metamaterials with geometric AI: a generative diffusion operates in compact latent space of cellular structures. *npj Metamaterials*, 1(1):5, 2025. doi: 10.1038/s44455-025-00005-6.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. doi: 10.18653/v1/2024.acl-long.662.

Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W. Pellegrini, Ralf S. Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*, 2019.

Jan-Hendrik Bastek and Dennis M. Kochmann. Inverse design of nonlinear mechanical metamaterials via video denoising diffusion models. *Nature Machine Intelligence*, 5(12):1466–1475, 2023. doi: 10.1038/s42256-023-00762-x.

Jan-Hendrik Bastek, Siddhant Kumar, Bastian Telgen, Raphaël N. Glaesener, and Dennis M. Kochmann. Inverting the structure–property map of truss metamaterials by deep learning. *Proceedings of the National Academy of Sciences of the United States of America*, 119(1):e2111505119, 2022. doi: 10.1073/pnas.2111505119.

Gabriel Cardoso, Yazid Janati el Idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *International Conference on Learning Representations*, 2024.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

G. R. Cowper. The shear coefficient in Timoshenko's beam theory. *Journal of Applied Mechanics*, 33(2):335–340, 1966. doi: 10.1115/1.3625046.

Peng Dai, Kai Sun, Xingzhao Yan, Otto L. Muskens, C. H. (Kees) de Groot, Xupeng Zhu, Yueqiang Hu, Huigao Duan, and Ruomeng Huang. Inverse design of structural color: finding multiple solutions via conditional generative adversarial networks. *Nanophotonics*, 11(13):3057–3069, 2022. doi: 10.1515/nanoph-2022-0095.

Bolei Deng, Ahmad Zareei, Xiaoxiao Ding, James C. Weaver, Chris H. Rycroft, and Katia Bertoldi. Inverse design of mechanical metamaterials with target nonlinear response via a neural accelerated evolution strategy. *Advanced Materials*, 34(41):2206238, 2022. doi: 10.1002/adma.202206238.

Francisco Dos Reis and Nikolaos Karathanasopoulos. Inverse metamaterial design combining genetic algorithms with asymptotic homogenization schemes. *International Journal of Solids and Structures*, 250:111702, 2022. doi: 10.1016/j.ijsolstr.2022.111702.

Francisco Dos Reis and Nikolaos Karathanasopoulos. Deep learning, deconvolutional neural network inverse design of strut-based lattice metamaterials. *Computational Materials Science*, 244:113258, 2024. doi: 10.1016/j.commatsci.2024.113258.

Ramón Frey, Michael R. Tucker, Mohamadreza Afrasiabi, and Markus Bambach. Multi-objective design of multi-material truss lattices utilizing graph neural networks. *Scientific Reports*, 15(1): 3187, 2025. doi: 10.1038/s41598-025-86812-3.

Raphaël N. Glaesener, Erik A. Träff, Bastian Telgen, Renato M. Canonica, and Dennis M. Kochmann. Continuum representation of nonlinear three-dimensional periodic truss networks by on-the-fly homogenization. *International Journal of Solids and Structures*, 206:101–113, 2020. doi: 10.1016/j.ijsolstr.2020.08.013.

Chan Soo Ha, Desheng Yao, Zhenpeng Xu, Chenang Liu, Han Liu, Daniel Elkins, Matthew Kile, Vikram Deshpande, Zhenyu Kong, Mathieu Bauchy, and Xiaoyu Zheng. Rapid inverse design of metamaterials based on prescribed mechanical behavior through machine learning. *Nature Communications*, 14(1):5765, 2023. doi: 10.1038/s41467-023-40854-1.

Amin Heyrani Nobari, Wei Chen, and Faez Ahmed. PcDGAN: A continuous conditional diverse generative adversarial network for inverse design. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 606–616. ACM, 2021. doi: 10.1145/3447548.3467414.

Namjung Kim, Dongseok Lee, Chanyoung Kim, Dosung Lee, and Youngjoon Hong. Simple arithmetic operation in latent space can generate a novel three-dimensional graph metamaterials. *npj Computational Materials*, 10(1):236, 2024. doi: 10.1038/s41524-024-01430-3.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. SetPO: Set-level policy optimization for diversity-preserving LLM reasoning, 2026a. URL https://arxiv.org/abs/2602.01062.

Yijie Li, Jiahao Xu, Ching-Chih Tsao, Lili Qiu, and Jingxian Wang. Physics-guided sequence-based generative framework for acoustic metamaterial inverse design, 2026b. URL https://arxiv.org/abs/2606.09266.

Jiaoxuan Lin, Bowen Zhang, Teng Ma, Hengji Dong, Anfeng Huang, Tianhao Zhao, Hongwei Yang, Weizhe Zhang, Changguo Wang, Hui He, and Yuanpeng Liu. TrussGPT: large language model-driven inverse design framework for truss metamaterials. *npj Computational Materials*, 12 (1):301, 2026. doi: 10.1038/s41524-026-02175-x.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization, 2026. URL https://arxiv.org/abs/2601.05242.

Wei Ma, Feng Cheng, Yihao Xu, Qinlong Wen, and Yongmin Liu. Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. *Advanced Materials*, 31(35):1901111, 2019. doi: 10.1002/adma.201901111.

Marco Maurizi, Chao Gao, and Filippo Berto. Inverse design of truss lattice materials with superior buckling resistance. *npj Computational Materials*, 8(1):247, 2022. doi: 10.1038/s41524-022-00938-w.

Marco Maurizi, Derek Xu, Yu-Tong Wang, Desheng Yao, David Hahn, Mourad Oudich, Anish Satpati, Mathieu Bauchy, Wei Wang, Yizhou Sun, Yun Jing, and Xiaoyu Rayne Zheng. Designing metamaterials with programmable nonlinear responses and geometric constraints in graph space. *Nature Machine Intelligence*, 7(7):1023–1036, 2025. doi: 10.1038/s42256-025-01067-x.

Mechanics and Materials Lab. ae108, 2020. URL https://doi.org/10.5905/ethz-1007-257.

Sushan Nakarmi, Nitin P. Daphalapurkar, Kwan-Soo Lee, Jihyeon Kim, Jeffery A. Leiding, Dana M. Dattelbaum, and Darby J. Luscher. Inverse design of cellular structures with the targeted nonlinear mechanical response. *Scientific Reports*, 16(1):3185, 2026. doi: 10.1038/s41598-025-33184-3.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Saurav Sharma, Satya K. Ammu, Prakash Thakolkaran, Jovana Jovanova, Kunal Masania, and Siddhant Kumar. Piezoelectric truss metamaterials: data-driven design and additive manufacturing. *npj Metamaterials*, 1(1):9, 2025. doi: 10.1038/s44455-025-00009-2.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler–lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. Optimizing language models for inference time objectives using reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59066–59085, 2025.

Than V. Tran, S. S. Nanthakumar, and Xiaoying Zhuang. Deep learning-based framework for the on-demand inverse design of metamaterials with arbitrary target band gap. *npj Artificial Intelligence*, 1(1):2, 2025. doi: 10.1038/s44387-025-00001-1.

Rohit Unni, Kan Yao, and Yuebing Zheng. Deep convolutional mixture density network for inverse design of layered photonic structures. *ACS Photonics*, 7(10):2703–2712, 2020. doi: 10.1021/acsphotonics.0c00630.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

Christian Walder and Deep Tejas Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5096.

Jun Wang, Wei (Wayne) Chen, Daicong Da, Mark Fuge, and Rahul Rai. IH-GAN: A conditional generative model for implicit surface-based inverse design of cellular structures. *Computer Methods in Applied Mechanics and Engineering*, 396:115060, 2022. doi: 10.1016/j.cma.2022.115060.

Ling-Qi Wang, Jing-Zhong Tong, Ken E. Evans, and Jiajia Shen. Performance-based inverse design of nonlinear mechanical metamaterials via enhanced conditional generative adversarial networks. *Computers & Structures*, 325:108183, 2026. doi: 10.1016/j.compstruc.2026.108183.

Tianyang Xue, Longdu Liu, Lin Lu, Paul Henderson, Pengbin Tang, Haochen Li, Jikai Liu, Haisen Zhao, Hao Peng, and Bernd Bickel. MIND: Microstructure INverse design with generative hybrid neural representation. In *Proceedings of the SIGGRAPH Conference Papers*, pp. 1–12. ACM, 2025. doi: 10.1145/3721238.3730682.

Jia-Qi Yang, YuCheng Xu, Kebin Fan, Jingbo Wu, Caihong Zhang, De-Chuan Zhan, Biao-Bing Jin, and Willie J. Padilla. Normalizing flows for efficient inverse design of thermophotovoltaic emitters. *ACS Photonics*, 10(4):1001–1011, 2023. doi: 10.1021/acsphotonics.2c01803.

Christopher Yeung, Ryan Tsai, Benjamin Pham, Brian King, Yusaku Kawagoe, David Ho, Julia Liang, Mark W. Knight, and Aaswath P. Raman. Global inverse design across multiple photonic structure classes using generative deep learning. *Advanced Optical Materials*, 9(20):2100548, 2021. doi: 10.1002/adom.202100548.

Wangzhi Zhan, Jianpeng Chen, Dongqi Fu, and Dawei Zhou. UniMate: A unified model for mechanical metamaterial generation, property prediction, and condition confirmation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74406–74421. PMLR, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Li Zheng, Konstantinos Karapiperis, Siddhant Kumar, and Dennis M. Kochmann. Unifying the design space and optimizing linear and nonlinear truss metamaterials by generative modeling. *Nature Communications*, 14(1):7563, 2023a. doi: 10.1038/s41467-023-42068-x.

Li Zheng, Siddhant Kumar, and Dennis M. Kochmann. Algebraic language models for inverse design of metamaterials via diffusion transformers. *Nature Machine Intelligence*, 8(4):628–640, 2026. doi: 10.1038/s42256-026-01218-8.

Xiaoyang Zheng, Ta-Te Chen, Xiaofeng Guo, Sadaki Samitsu, and Ikumu Watanabe. Controllable inverse design of auxetic metamaterials using deep learning. *Materials & Design*, 211:110178, 2021. doi: 10.1016/j.matdes.2021.110178.

Xiaoyang Zheng, Ta-Te Chen, Xiaoyu Jiang, Masanobu Naito, and Ikumu Watanabe. Deep-learning-based inverse design of three-dimensional architected cellular materials with the target porosity and stiffness using voxelized voronoi lattices. *Science and Technology of Advanced Materials*, 24 (1):2157682, 2023b. doi: 10.1080/14686996.2022.2157682.

APPENDIX

A EOTR REPRESENTATION AND DATASET CONSTRUCTION

A.1 CANONICAL REPRESENTATION AND SOURCE INTEGRATION

We construct the TrussOrbit dataset by converting Pmmm-compatible structures from UniTruss, GraphMetaMat, TrussGPT, and PiezoTruss into the common **Edge Orbit Truss Representation (EOTR)** (Zheng et al., 2023a; Maurizi et al., 2025; Lin et al., 2026; Sharma et al., 2025). TrussGPT supplies separate property and stress–strain collections, yielding five input collections from four published sources. Table 3 accounts for each input collection: validity filtering precedes the within-source counts of unique and duplicate records.

Table 3: Source collections before global deduplication.

Source	Observed	Valid	Invalid	Unique	Duplicates
UniTruss	960,244	960,244	0	960,195	49
GraphMetaMat	900,000	897,006	2,994	894,808	2,198
TrussGPT property	248,780	243,657	5,123	243,447	210
TrussGPT response	298,941	298,941	0	298,603	338
PiezoTruss DS2	965,685	962,471	3,214	960,195	2,276

We then deduplicate identical EOTR sequences across all sources. This reduces 3,362,319 valid input records to 1,492,505 canonical structures, a 55.61% reduction. Four rounds of Weisfeiler–Lehman refinement (Shervashidze et al., 2011) define the WL4 topology signature used to group the structures into 452,791 families. We assign each family to one split, so training, validation, and test are disjoint under this signature. Table 4 reports the resulting split sizes and sequence lengths.

Table 4: Dataset splits and EOTR sequence lengths.

Split	Structures	Mean length	Median	P95	Max
Train	1,336,666	74.433	75	98	146
Validation	86,577	75.372	75	97	130
Test	69,262	75.075	76	99	137
All	1,492,505	74.517	75	98	146

EOTR stores each node-orbit representative $\bar{v}_i$ as three coordinate tokens in (x, y, z) order. Edge-orbit record $(U_{i_m}, \mathrm{VGH}_{j_m,h_m})$ specifies the strut from $\bar{v}_{i_m}$ to $g_{h_m}(\bar{v}_{j_m})$; applying the eight Pmmm operations recovers its full edge orbit. Here i_m and j_m index the source and target node-orbit representatives, and h_m selects the relative symmetry operation. U_{i_m} is one source-index token, while VGH_{j_m,h_m} combines the target index and operation in one token, so each edge orbit uses two tokens. The token COORD_j has grid index $j \in \{0, \ldots, 1024\}$, numeric token ID $4+j$, and decoded coordinate value $j/1024$. All axes share these 1,025 tokens; position within the tuple identifies the axis. The midpoint $j = 512$ represents $1/2$ exactly and remains fixed under reflection across the coordinate midplane. The operations $g_0, \ldots, g_7$ follow the order $I, R_x, R_y, R_{xy}, R_z, R_{xz}, R_{yz}, R_{xyz}$. Here I is the identity, R_x, R_y, and R_z reflect across the corresponding coordinate midplanes, and combined subscripts denote compositions. Canonically ordered node and edge orbits give the stored length

$$L_{\mathrm{EOTR}}(\mathcal{T}) = 3N_v + 2N_e + 3. \tag{10}$$

Here N_v and N_e count node and edge orbits. The three additional tokens are SOS, EDGES, and EOS. Table 5 specifies the 2,469-token vocabulary: four special tokens, 1,025 coordinate tokens, 160 source-index tokens, and 160×8 joint target-operation tokens. Stored lengths exclude the continuous stiffness prefix; the codec accepts up to 1,024 discrete tokens, and the observed dataset maximum is 146. In training objectives, the scored length excludes the externally supplied SOS, as defined in Section B.1.

The following procedure specifies the canonical ordering used by the codec. Input coordinates are finite and lie in the source unit domain; an octant input is expanded into the full cell before

quantization. Coincident symmetry copies are merged by exact coordinate equality (zero tolerance, with signed zeros identified). Distinct full-cell nodes that collide after quantization are rejected. All sorting and minima below are lexicographic, and rint rounds to the nearest integer with ties to even. On integer coordinates, a reflection maps q_d to $M - q_d$. Operation IDs use the order defined above, so composition is bitwise XOR ($\oplus$).

Algorithm 1 EOTR canonicalization

Require: Coordinates X, undirected edges E, source mode; $M = 1024$.
  if octant input **then**
    Expand (X, E) using $x \mapsto (s \odot x + \mathbf{1})/2$, $s \in \{-1, 1\}^3$.
    Merge exactly coincident expanded nodes and remap edge endpoints.
  Replace E by its set of unordered endpoint pairs; reject self-loops.
  Quantize each full-cell coordinate: $q_v \leftarrow \mathrm{rint}(Mx_v)$.
  Reject duplicate quantized coordinates; require node and edge closure under all g_h.
  For each point q, set $r(q)_d \leftarrow \min(q_d, M - q_d)$ for $d \in \{x, y, z\}$.
  $R \leftarrow \mathrm{sort}(\mathrm{unique}\{r(q)\})$; index representatives from 1.
  For each point q, let $i(q)$ be the index satisfying $R_{i(q)} = r(q)$.
  $A(q) \leftarrow \{h \in \{0, \ldots, 7\} : g_h(R_{i(q)}) = q\}$; retain every matching operation.
  Initialize the edge-key set $K \leftarrow \varnothing$.
  for all edges $\{u, v\} \in E$, with $(p, q) = (q_u, q_v)$ **do**
    $C \leftarrow \{(i(p), i(q), a \oplus b) : a \in A(p),\ b \in A(q)\}$.
    $C \leftarrow C \cup \{(j, i, h) : (i, j, h) \in C\}$; $K \leftarrow K \cup \{\min C\}$.
  $K \leftarrow \mathrm{sort}(K)$; each distinct edge-orbit key now occurs once.
  Require expanded orbits to reconstruct the graph and pass codec size and topology checks.
  $t \leftarrow [\mathrm{SOS}] \,\|\, [\mathrm{COORD}_{R_{i,x}}, \mathrm{COORD}_{R_{i,y}}, \mathrm{COORD}_{R_{i,z}}]_{i=1}^{|R|} \,\|\, [\mathrm{EDGES}]$.
  For (i, j, h) in K order, append $[U_i, \mathrm{VGH}_{j,h}]$ to t.
  Append EOS; reject sequences exceeding the length limit; **return** t.

Figure 5 illustrates the node orbits and edge-orbit records for two representative nodes. A node on a reflection plane can have several operation IDs in $A(q)$; enumerating all of them and both endpoint orders before taking the minimum gives one edge key. Small codec checks cover generic nodes, reflection-plane nodes, plane intersections, and the cell center: 40 random node relabelings, edge permutations, and endpoint reversals per test graph preserve its sequence, and decoding followed by re-encoding reproduces the same sequence.

Table 5: EOTR token vocabulary.

Sequence part	Token	Token IDs	Count	Description
Special	PAD	0	1	Padding
	SOS	1	1	Sequence start
	EDGES	2	1	Node/edge delimiter
	EOS	3	1	Sequence end
Node geometry	COORD_j	4–1028	1,025	Grid index $j = 0, \ldots, 1024$; ID $4 + j$; value $j/1024$. Shared by x, y, and z.
Edge orbits	U_i	1029–1188	160	Source node-orbit index $i = 1, \ldots, 160$
	$\mathrm{VGH}_{j,h}$	1189–2468	1,280	Target node-orbit index $j = 1, \ldots, 160$ and operation $h = 0, \ldots, 7$; ID $1189 + 8(j - 1) + h$

A.2 Topology complexity and stiffness coverage

Figure 7 illustrates the range of connectivity patterns in the dataset.

We characterize each connected full-cell graph by its node count $|V|$, edge count $|\mathcal{E}|$, cycle rank $|\mathcal{E}| - |V| + 1$, and mean degree $\frac{2|\mathcal{E}|}{|V|}$. Ten non-overlapping integer edge-count ranges span 24–260 struts. Figure 8 compares their frequency and conditional topology statistics across splits. A missing bar in a topology-statistic panel denotes an empty split/bin combination.

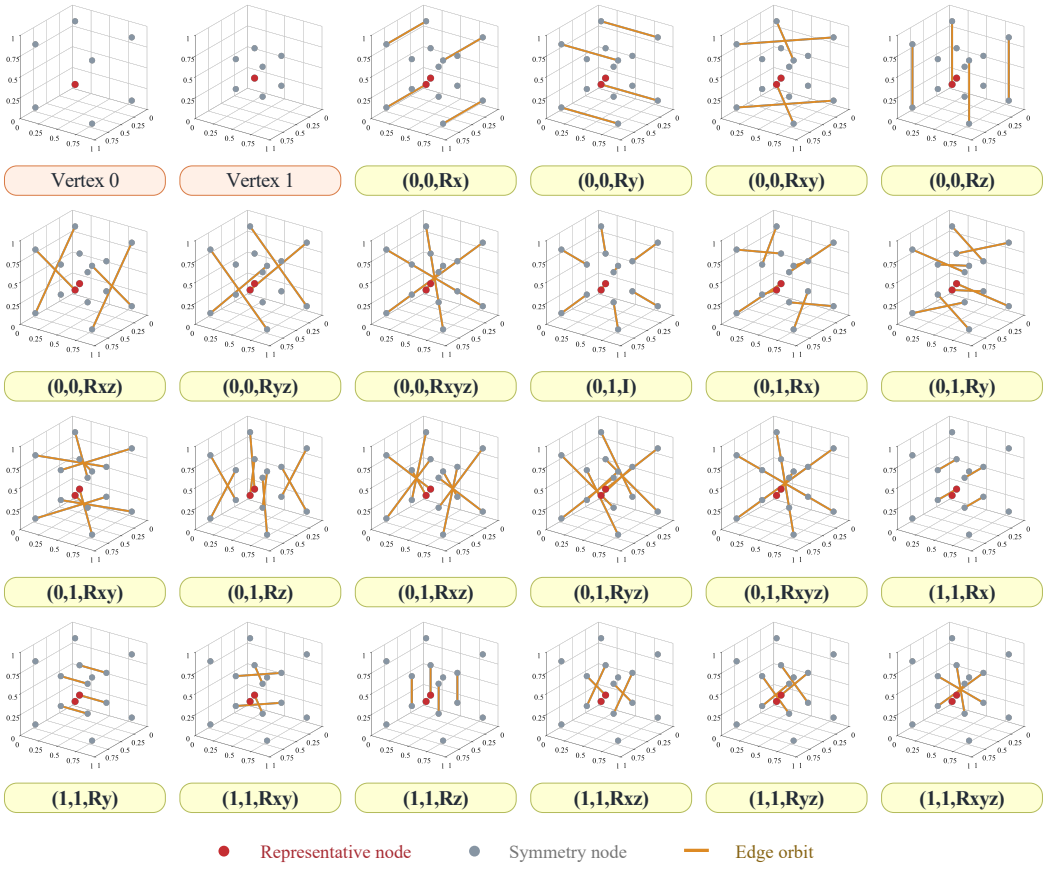


Figure 5: Node and edge orbits under Pmmm symmetry. Red and gray points denote representative and symmetry-related nodes. Orange segments show the edge orbit encoded by U_i and $\mathrm{VGH}_{j,h}$, where h selects the relative symmetry operation.

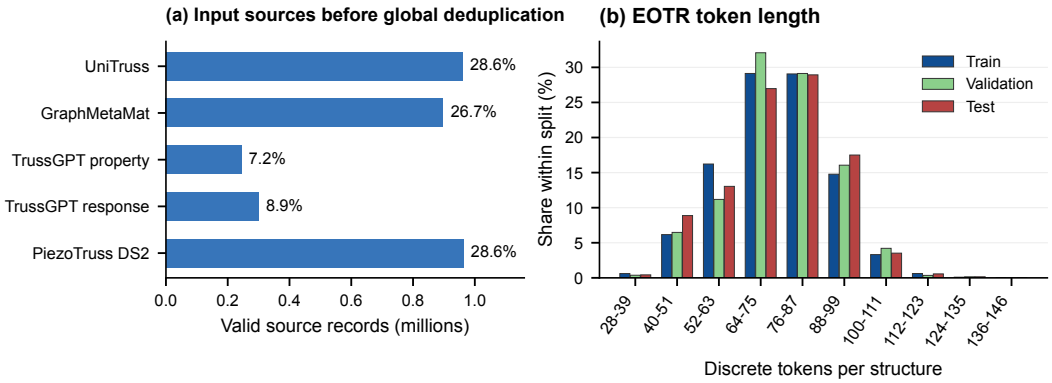


Figure 6: Source records and EOTR lengths. (a) Shares refer to valid source records before global deduplication; overlapping collections are counted at their inputs. (b) Each histogram is normalized by its own split size.

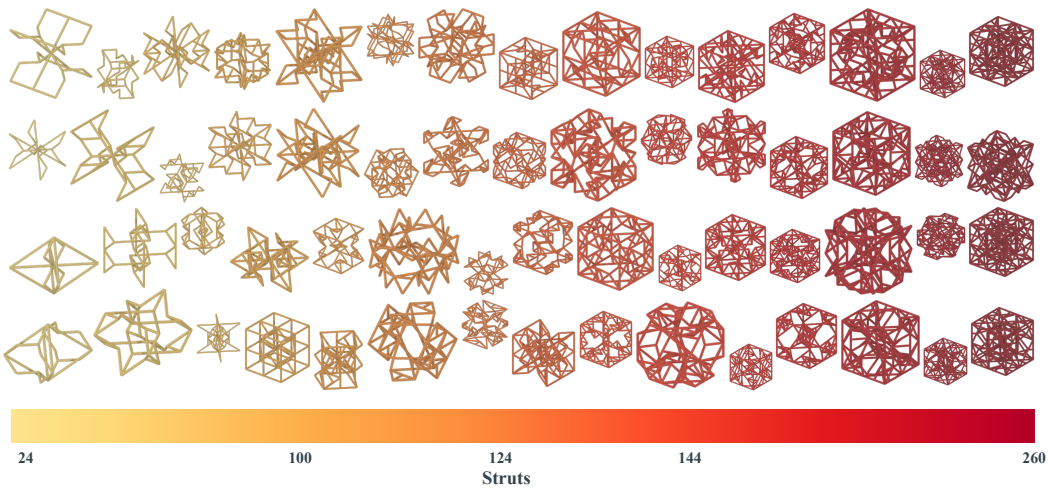


Figure 7: Examples from the TrussOrbit dataset, colored by full-cell strut count (24–260), illustrating sparse and dense connectivity patterns.

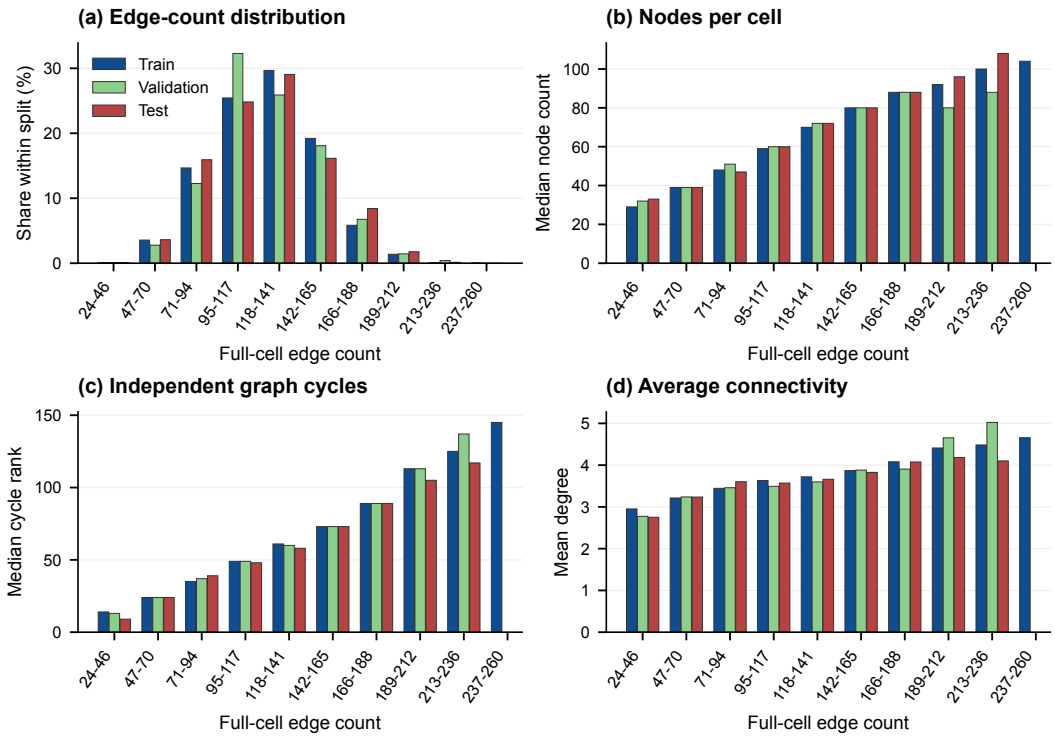
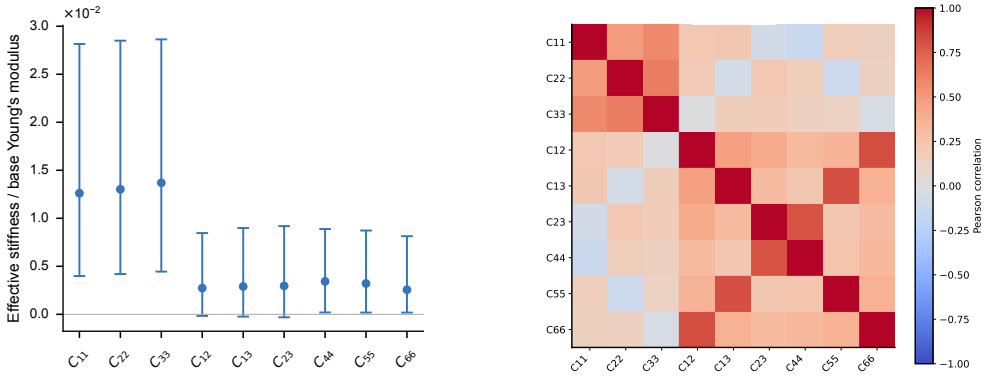


Figure 8: Topology complexity by full-cell edge count. (a) Within-split frequency. (b) Median node count. (c) Median cycle rank. (d) Mean of the per-structure mean degree. All four panels use the same edge-count bins and split colors.

All 1,492,505 canonical records have labels from the common ae108 pipeline in Section C. We use the main-paper component order $(C_{11}, C_{22}, C_{33}, C_{12}, C_{13}, C_{23}, C_{44}, C_{55}, C_{66})$. Figure 9 summarizes the full-corpus stiffness distributions and correlations. Conditioning means and standard deviations, min–max ranges for evaluation, and 5th–95th percentile widths for reinforcement-learning rewards are computed from the training split and frozen before use.



(a) Stiffness distribution

(b) Component correlations

Figure 9: Stiffness coverage. (a) Median and 5th–95th percentile intervals across the full corpus. (b) Correlations between stiffness components.

A.3 Dataset scale and topology diversity

Table 6 separates dataset size from topology coverage in published three-dimensional truss collections. Bastek et al. (2022) generate geometric variants of source-defined topologies, while UniTruss combines node and edge perturbations with superposition (Zheng et al., 2023a). The TrussOrbit dataset counts canonical structures after exact-EOTR deduplication and groups them by the WL4 signature used for splitting.

Table 6: Scale and topology coverage of physics-labeled 3D truss datasets. Published topology counts follow each source's definition; TrussOrbit counts WL4 families after exact-EOTR deduplication.

Dataset	Records	Topology count	Stiffness labels
Bastek et al. (2022)	3,000,000	262	21 components
UniTruss (Zheng et al., 2023a)	965,736	2,848	9 components
TrussOrbit dataset	1,492,505	452,791	9 components

Geometric variants contribute separate records, whereas topology counts group structures by connectivity. The source-specific topology definitions and our WL4 grouping differ, so the counts describe each collection under its stated convention. A direct topology ranking requires a shared graph construction and equivalence criterion.

B Complete Derivation of SMCO

Setwise Marginal Credit Optimization (SMCO) assigns physical reward credit through fixed-cardinality null-slot comparisons. This section defines the sequence probabilities and baseline objectives, derives the raw marginal-credit identity, and specifies the normalized GSPO training surrogate.

B.1 Notation and sequence probabilities

For target stiffness $\mathbf{c}^\star \in \mathbb{R}^9$, let $\mathbf{s}_i = (s_{i,1}, \ldots, s_{i,L_i})$ be the scored EOTR output following a supplied SOS. When decoding succeeds, the sequence specifies truss $\mathcal{T}_i$. Its probability is $p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star) =$

$\prod_{t=1}^{L_i} p_\theta(s_{i,t} \mid \mathbf{c}^\star, \mathbf{s}_{i,<t})$. The scored length L_i includes EOS and excludes padding, continuous prefixes, and SOS; a complete stored sequence therefore has $L_i = L_{\mathrm{EOTR}}(\mathcal{T}_i) - 1$. For one target, $\mathcal{S} = ((\mathbf{s}_i, \mathcal{T}_i))_{i=1}^k$ denotes the rollout group, with an absent structure for a decoding failure. Each sampled sequence retains its slot, including repetitions and failures. The reward is invariant to slot order.

The policy-gradient identity below uses samples from p_θ. The clipped objectives use samples from the frozen rollout policy $p_{\theta_{\mathrm{old}}}$. Rewards, old log probabilities, and advantages are held fixed during each policy-update phase. We define the token ratio and GSPO sequence ratio (Shao et al., 2024; Zheng et al., 2025) by

$$\rho_{i,t}(\theta) = \frac{p_\theta(s_{i,t} \mid \mathbf{c}^\star, \mathbf{s}_{i,<t})}{p_{\theta_{\mathrm{old}}}(s_{i,t} \mid \mathbf{c}^\star, \mathbf{s}_{i,<t})}, \tag{11}$$

$$\eta_i(\theta) = \left[\frac{p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star)}{p_{\theta_{\mathrm{old}}}(\mathbf{s}_i \mid \mathbf{c}^\star)}\right]^{1/L_i} = \exp\left[\frac{1}{L_i}\sum_{t=1}^{L_i} \log \rho_{i,t}(\theta)\right]. \tag{12}$$

For scalar ratio z and scalar advantage A, define the clipping operator

$$\mathcal{C}(z, A) = \min\{zA, \mathrm{clip}(z, 1-\epsilon_{\mathrm{l}}, 1+\epsilon_{\mathrm{h}})A\}. \tag{13}$$

Setting $\epsilon_{\mathrm{l}} = \epsilon_{\mathrm{h}}$ recovers symmetric clipping. The nonnegative clip widths are method-specific hyperparameters; $\mathrm{clip}(z, a, b)$ truncates z to the interval $[a, b]$.

For a vector $\mathbf{z} \in \mathbb{R}^k$, define $\mu(\mathbf{z}) = k^{-1}\sum_i z_i$, $\sigma(\mathbf{z}) = [k^{-1}\sum_i (z_i - \mu(\mathbf{z}))^2]^{1/2}$, and

$$\mathcal{N}_i(\mathbf{z}) = \frac{z_i - \mu(\mathbf{z})}{\sigma(\mathbf{z}) + \varepsilon_A}, \qquad \varepsilon_A > 0. \tag{14}$$

A constant vector gives zero advantage.

B.2 Baseline policy objectives in the same notation

Let $\mathcal{D}_{\mathrm{target}}$ denote the training-target distribution and $\beta \geq 0$ the KL-regularization weight. Expectations in clipped objectives average over targets and groups sampled from the old policy. For a candidate that passes parsing, structural checks, and FEM, let $v_i^{\mathrm{val}} = 1$ and $q_i = \exp(-\mathrm{NRMSE}_i^{\mathrm{RL}}/\tau_u)$, with $\mathrm{NRMSE}_i^{\mathrm{RL}}$ and τ_u defined in Appendix B; set both rewards to zero for a failed candidate. The scalar task reward used by RLOO, GRPO, GSPO, and SetPO is $r_i = q_i + \lambda_{\mathrm{val}} v_i^{\mathrm{val}}$, with $\lambda_{\mathrm{val}} = 0.5$, and $\mathbf{r} = (r_1, \ldots, r_k)$ collects the group rewards. When enabled, KL regularization uses a frozen reference policy p_{ref}. These baselines share the task reward, validity checks, and test-time evaluation protocol. Their training group sizes, sampling settings, and optimizers are specified in Table 9.

B.2.1 GRPO

GRPO uses the group-relative outcome advantage $\bar{A}_i = \mathcal{N}_i(\mathbf{r})$ for every scored token (Shao et al., 2024). We follow the group-relative estimator in the DeepSeekMath formulation and its official implementation references DeepSeekMath and veRL. With $\ell_{i,t} = \log p_\theta(s_{i,t} \mid \mathbf{c}^\star, \mathbf{s}_{i,<t}) - \log p_{\mathrm{ref}}(s_{i,t} \mid \mathbf{c}^\star, \mathbf{s}_{i,<t})$, define the sampled KL term

$$\kappa_{i,t} = e^{-\ell_{i,t}} + \ell_{i,t} - 1. \tag{15}$$

For the official veRL GRPO recipe used here, $\beta = 0.001$ is applied through the low-variance KL loss, and the token clip width is 0.2; Table 9 records the remaining method settings. The token-level clipped objective is

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{k}\sum_i \frac{1}{L_i}\sum_t \left\{\mathcal{C}(\rho_{i,t}(\theta), \bar{A}_i) - \beta\kappa_{i,t}\right\}\right]. \tag{16}$$

The KL penalty enters the objective separately from the group-normalized reward. Setting $\beta = 0$ removes reference regularization. Implementation computes a ratio and clipping decision for every token before averaging within each sequence and then across the group.

B.2.2 GSPO

GSPO retains $\bar{A}_i = \mathcal{N}_i(\mathbf{r})$ and replaces token ratios by Equation 12 (Zheng et al., 2025). We use the sequence-level policy-loss formulation in the official veRL GSPO implementation, with lower and upper clip widths 3×10^{-4} and 4×10^{-4} and no KL loss; Table 9 gives its rollout and optimizer settings:

$$\mathcal{J}_{\mathrm{GSPO}}(\theta) = \mathbb{E}\left[\frac{1}{k}\sum_i \mathcal{C}(\eta_i(\theta), \bar{A}_i)\right]. \tag{17}$$

The sequence log-probability difference is averaged over its scored tokens before exponentiation. Each sequence then contributes one clipped scalar term, with no additional length division outside η_i. A separately enabled reference penalty can be added to this policy term.

B.2.3 RLOO

The REINFORCE leave-one-out estimator treats a complete sequence as one action (Ahmadian et al., 2024). Our implementation follows the sequence-reward and leave-one-out construction documented by the official TRL RLOO trainer. We use two samples per target, learning rate 10^{-6}, and KL coefficient $\beta = 0.05$; Table 9 records the training duration. The estimator is unclipped, as in the cited formulation. At the on-policy sampling point, define the KL-shaped return and its baseline by

$$\mathcal{R}_i = r_i - \beta \log \frac{p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star)}{p_{\mathrm{ref}}(\mathbf{s}_i \mid \mathbf{c}^\star)}, \qquad A_i^{\mathrm{RLOO}} = \mathcal{R}_i - \frac{1}{k-1}\sum_{j\neq i}\mathcal{R}_j, \quad k \geq 2. \tag{18}$$

The other sequences are independent of sequence i conditional on the target, so their mean is a valid score-function baseline. The update is

$$\widehat{\nabla_\theta \mathcal{J}}_{\mathrm{RLOO}} = \frac{1}{k}\sum_i A_i^{\mathrm{RLOO}} \sum_{t=1}^{L_i} \nabla_\theta \log p_\theta(s_{i,t} \mid \mathbf{c}^\star, \mathbf{s}_{i,<t}). \tag{19}$$

It sums token scores without standard-deviation normalization or length division. Writing $\overline{\mathcal{R}} = k^{-1}\sum_i \mathcal{R}_i$ for the group mean, $A_i^{\mathrm{RLOO}} = \frac{k(\mathcal{R}_i - \overline{\mathcal{R}})}{k-1}$, so the advantage can be computed from one group sum. The update is on-policy and unclipped.

B.2.4 SetPO

SetPO augments a normalized task advantage with marginal set diversity (Li et al., 2026a). We use the GRPO-based SetPO variant and its leave-one-out semantic-diversity construction, following the official SetPO implementation. For a symmetric similarity $K_{ij} \in [0, 1]$, its empirical diversity is

$$m_i = \frac{1}{k-1}\sum_{j\neq i} K_{ij}, \qquad \psi(z) = -\log(1+z), \qquad D(\mathcal{S}) = \frac{1}{k}\sum_i \psi(m_i). \tag{20}$$

Deleting candidate i changes both the candidate count and each remaining local-mass denominator:

$$m_j^{(-i)} = \frac{\sum_{\ell\neq i,j} K_{j\ell}}{k-2}, \qquad d_i = D(\mathcal{S}) - \frac{1}{k-1}\sum_{j\neq i}\psi(m_j^{(-i)}), \quad k \geq 3. \tag{21}$$

With diversity weight $\lambda_{\mathrm{SetPO}} = 0.05$, the augmented advantage is $A_i^{\mathrm{SetPO}} = \bar{A}_i + \lambda_{\mathrm{SetPO}} d_i$. Our SetPO comparison uses its GRPO-based formulation: this advantage enters Equation 16 without further standardization. For nonzero sequence embeddings $\mathbf{e}_i$ from Qwen3-Embedding-0.6B, the clipped cosine similarity is (Li et al., 2026a)

$$K_{ij} = \mathrm{clip}\left(\frac{\mathbf{e}_i^\top \mathbf{e}_j}{\|\mathbf{e}_i\|_2 \|\mathbf{e}_j\|_2}, 0, 1\right). \tag{22}$$

Pairwise similarities and row sums suffice to compute all deletion margins in $O(k^2)$ operations. The following section derives SMCO by applying marginal credit to property and validity rewards.

B.3 SETWISE MARGINAL CREDIT OPTIMIZATION

Inspired by SetPO's marginal set-credit principle (Li et al., 2026a), SMCO combines physical set rewards, fixed-cardinality credit, and group-normalized advantages. The rewards measure stiffness matching and binary validity; null-slot comparisons assign credit to individual candidates at unchanged cardinality; and normalized credit channels supply the learning signal for a GSPO sequence-level update (Zheng et al., 2025). The derivation below first establishes raw marginal credit, then distinguishes it from the normalized, clipped training surrogate.

B.3.1 SET REWARDS AND EXACT MARGINAL CONTRIBUTIONS

As in the main paper, let $v_i^{\mathrm{val}} \in \{0, 1\}$ equal one if candidate i passes all validity checks and zero otherwise. The checks cover parsing, legal nodes and struts, periodic connectivity after Pmmm expansion, and successful FEM evaluation.

For a successful candidate, FEM returns $\widehat{\mathbf{c}}_i = \mathbf{c}(\mathcal{T}_i)$. We compute $\mathrm{NRMSE}_i^{\mathrm{RL}}$ against the target $\mathbf{c}^\star$, normalizing each component error by the width of that component's training-set 5th–95th percentile interval for training stability. Evaluation uses the min–max ranges in Section D. We set $q_i = \exp(-\mathrm{NRMSE}_i^{\mathrm{RL}}/\tau_u)$ when $v_i^{\mathrm{val}} = 1$, with temperature $\tau_u > 0$, and $q_i = 0$ otherwise.

For smoothing temperature $\tau_b > 0$, the two set rewards in the main paper are

$$R_{\mathrm{fit}}(\mathcal{S}) = \tau_b \log\left[\frac{1}{k}\sum_{j=1}^{k} e^{q_j/\tau_b}\right], \qquad R_{\mathrm{val}}(\mathcal{S}) = \frac{1}{k}\sum_{j=1}^{k} v_j^{\mathrm{val}}. \tag{23}$$

For $q_{\max} = \max_j q_j$, the log-sum-exp bounds give

$$q_{\max} - \tau_b \log k \le R_{\mathrm{fit}}(\mathcal{S}) \le q_{\max}, \qquad 0 \le R_{\mathrm{fit}}(\mathcal{S}) \le 1. \tag{24}$$

Thus $R_{\mathrm{fit}} \to q_{\max}$ as $\tau_b \to 0$. The finite-temperature derivative distributes sensitivity across the candidates:

$$\frac{\partial R_{\mathrm{fit}}}{\partial q_i} = \frac{e^{q_i/\tau_b}}{\sum_{j=1}^{k} e^{q_j/\tau_b}}. \tag{25}$$

This reward targets multi-sample quality (Tang et al., 2025; Walder & Karkhanis, 2025).

Replace slot i by a null candidate with $q_i = v_i^{\mathrm{val}} = 0$ to obtain $\mathcal{S}^{0,i}$. This keeps the group size k fixed. The null slot contributes $e^0 = 1$ to the exponential sum. Writing $Z = \sum_j e^{q_j/\tau_b}$ gives the exact marginal rewards

$$\Delta_i^{(\mathrm{fit})} = R_{\mathrm{fit}}(\mathcal{S}) - R_{\mathrm{fit}}(\mathcal{S}^{0,i}) = \tau_b \log \frac{Z}{Z - e^{q_i/\tau_b} + 1}, \tag{26}$$

$$\Delta_i^{(\mathrm{val})} = R_{\mathrm{val}}(\mathcal{S}) - R_{\mathrm{val}}(\mathcal{S}^{0,i}) = \frac{v_i^{\mathrm{val}}}{k}. \tag{27}$$

Both margins are nonnegative. A slot with zero score has zero margin in that channel, and equal $(q_i, v_i^{\mathrm{val}})$ pairs receive equal margins, including repeated structures. These properties follow from the physical rewards and fixed null baseline; they do not require pairwise sequence similarities.

B.3.2 ON-POLICY CREDIT IDENTITY

For fixed $\mathbf{c}^\star$, independent sampling gives $p_\theta(\mathcal{S} \mid \mathbf{c}^\star) = \prod_i p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star)$. Assuming differentiation and expectation can be interchanged, the score-function identity yields

$$\nabla_\theta \mathbb{E}_{\mathcal{S}\sim p_\theta^k}[R_h(\mathcal{S})] = \mathbb{E}\left[R_h(\mathcal{S}) \sum_{i=1}^{k} \nabla_\theta \log p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star)\right], \quad h \in \{\mathrm{fit}, \mathrm{val}\}. \tag{28}$$

Let $\mathcal{S}_{-i}$ denote the other $k-1$ rollout slots. Conditional on these sequences, $R_h(\mathcal{S}^{0,i})$ is independent of $\mathbf{s}_i$. Its score expectation vanishes:

$$\begin{aligned}&\mathbb{E}\big[R_h(\mathcal{S}^{0,i})\nabla_\theta \log p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star) \mid \mathcal{S}_{-i}\big] \\ &\quad = R_h(\mathcal{S}^{0,i}) \sum_{\mathbf{s}_i} p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star)\nabla_\theta \log p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star) = 0.\end{aligned} \tag{29}$$

Subtracting these zero-expectation terms establishes

$$\boxed{\nabla_\theta \mathbb{E}[R_h(\mathcal{S})] = \mathbb{E}\left[\sum_{i=1}^{k} \Delta_i^{(h)} \nabla_\theta \log p_\theta(\mathbf{s}_i \mid \mathbf{c}^\star)\right].} \tag{30}$$

The null-slot reward serves as a candidate-independent baseline, analogous to leave-one-out baselines (Ahmadian et al., 2024; Li et al., 2026a). The identity holds for raw on-policy margins and extends linearly to fixed weighted sums of R_{fit} and R_{val}.

B.3.3 Normalized advantages and the GSPO surrogate

To control the relative influence of credit channels with different scales and obtain a centered sequence-level learning signal, we independently standardize the two margin channels, then combine and standardize them within the target:

$$A_i^{(h)} = \mathcal{N}_i(\boldsymbol{\Delta}^{(h)}), \qquad \widetilde{A}_i = A_i^{(\text{fit})} + \lambda_{\text{val}} A_i^{(\text{val})}, \qquad \widehat{A}_i^{\text{set}} = \mathcal{N}_i(\widetilde{\mathbf{A}}). \tag{31}$$

Following Liu et al. (2026), we normalize reward channels separately. Our second normalization is within each target group; GDPO's final normalization is over the batch. The nonnegative validity weight λ_{val} acts on standardized channels.

GDPO informs the construction of the advantages above, while GSPO supplies the length-normalized sequence ratio and clipped policy update (Zheng et al., 2025). Substituting Equation 31 into the GSPO objective gives

$$\mathcal{J}_{\text{SMCO}}(\theta) = \mathbb{E}_{\substack{\mathbf{c}^\star \sim \mathcal{D}_{\text{target}} \\ \mathcal{S} \sim p_{\theta_{\text{old}}}^k}} \left[\frac{1}{k} \sum_{i=1}^{k} \mathcal{C}\left(\eta_i(\theta), \widehat{A}_i^{\text{set}}\right)\right]. \tag{32}$$

At $\theta = \theta_{\text{old}}$, with clipping inactive,

$$\nabla_\theta \mathcal{J}_{\text{SMCO}} = \mathbb{E}\left[\frac{1}{k} \sum_i \frac{\widehat{A}_i^{\text{set}}}{L_i} \sum_{t=1}^{L_i} \nabla_\theta \log p_\theta(s_{i,t} \mid \mathbf{c}^\star, \mathbf{s}_{i,<t})\right]. \tag{33}$$

Group normalization changes the credit scale, length normalization introduces $1/L_i$, and clipping limits subsequent updates. Equation 32 is a normalized, clipped surrogate of the raw-reward objective. All tokens in a sequence share its advantage and sequence-ratio clipping decision (Zheng et al., 2025).

Implementation. For every target, sample k sequences and store their non-padding masks and old token log probabilities. Decode each candidate, run the validity checks, and evaluate eligible structures once with ae108. Store v_i^{val}, compute $\text{NRMSE}_i^{\text{RL}}$ for successful candidates, and set failed candidates to $q_i = 0$. Compute Equation 26 as the difference of two log-sum-exp evaluations, replacing the selected logit by zero in the second, to avoid exponential overflow and subtractive cancellation. Normalize the margin channels, freeze the resulting advantages, and maximize Equation 32. FEM values are reused for all counterfactuals. Invalid rollouts retain their slots, and all-equal groups contribute zero policy advantage. Sampling probabilities used in the ratio must match the rollout distribution, including any temperature or support restriction.

C The ae108 Solver

This section specifies the common physical evaluator used for dataset labeling and generated candidates. Under Pmmm symmetry and linear elasticity, the effective stiffness matrix has the orthotropic form (Zheng et al., 2023a)

$$\mathbf{C}(\mathcal{T}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix}. \tag{34}$$

C.1 BEAM MODEL AND DENSITY CONTROL

We follow the linear computational-homogenization setting of UniTruss (Zheng et al., 2023a), implemented with ae108 (Mechanics and Materials Lab, 2020), and the periodic-network formulation of Glaesener et al. (2020). The representative cell is $[0,1]^3$, with volume $V_{\text{cell}} = 1$. Struts are circular, linear-elastic Timoshenko beams with base Young's modulus $E = 1$, Poisson ratio $\nu = 0.3$, and shear modulus $G = \frac{E}{2(1+\nu)}$. Their common radius r is adjusted independently for each structure to impose relative density $\rho = 0.15$.

Boundary struts can appear as multiple translated copies in the closed-cell graph. Let μ_e be the multiplicity of strut e under lattice translation and let $w_e = \frac{1}{\mu_e}$. With strut length ℓ_e, density and radius are evaluated using the same periodic weights:

$$\rho = \frac{\pi r^2}{V_{\text{cell}}} \sum_{e \in \mathcal{E}} w_e \ell_e, \qquad r = \sqrt{\frac{\rho V_{\text{cell}}}{\pi \sum_{e \in \mathcal{E}} w_e \ell_e}}. \tag{35}$$

The circular section has cross-sectional area $A = \pi r^2$, second area moments $I_y = I_z = \frac{\pi r^4}{4}$, and torsional moment $J = \frac{\pi r^4}{2}$. For circular sections, Cowper (1966) gives the shear coefficient $k_{\text{sh}} = \frac{6(1+\nu)}{7+6\nu}$.

C.2 PERIODIC EQUILIBRIUM AND EFFECTIVE STIFFNESS

Each beam node has translation and rotation degrees of freedom. For macroscopic infinitesimal strain $\bar{\boldsymbol{\epsilon}}$, write the nodal displacement as $\mathbf{u}(\mathbf{x}) = \bar{\boldsymbol{\epsilon}}\mathbf{x} + \tilde{\mathbf{u}}(\mathbf{x})$. Lattice-equivalent boundary points obey

$$\mathbf{u}(\mathbf{x}^+) - \mathbf{u}(\mathbf{x}^-) = \bar{\boldsymbol{\epsilon}}(\mathbf{x}^+ - \mathbf{x}^-), \qquad \boldsymbol{\varphi}(\mathbf{x}^+) = \boldsymbol{\varphi}(\mathbf{x}^-). \tag{36}$$

Here $\mathbf{x}^+$ and $\mathbf{x}^-$ are lattice-equivalent positions, and $\boldsymbol{\varphi}$ denotes the nodal rotation vector. The fluctuation $\tilde{\mathbf{u}}$ is periodic, and one representative node is fixed in translation to remove rigid translation. Solving the constrained beam equilibrium minimizes the weighted cell energy. Let $\mathbf{z}$ collect the independent fluctuation and rotation degrees of freedom, and define the engineering-strain vector

$$\mathbf{e} = (\bar{\epsilon}_{11}, \bar{\epsilon}_{22}, \bar{\epsilon}_{33}, 2\bar{\epsilon}_{23}, 2\bar{\epsilon}_{13}, 2\bar{\epsilon}_{12})^{\mathsf{T}}. \tag{37}$$

For linear beams, the constrained energy is quadratic. Partitioning its Hessian with respect to $\mathbf{e}$ and $\mathbf{z}$ gives

$$W^\star(\mathbf{e}) = \min_{\mathbf{z}} \frac{1}{2V_{\text{cell}}} \begin{bmatrix} \mathbf{e} \\ \mathbf{z} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} H_{ee} & H_{ez} \\ H_{ze} & H_{zz} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{z} \end{bmatrix}, \qquad \mathbf{C} = \frac{H_{ee} - H_{ez} H_{zz}^{-1} H_{ze}}{V_{\text{cell}}}. \tag{38}$$

The blocks H_{ee}, H_{ez}, H_{ze}, and H_{zz} are the strain, cross, and fluctuation blocks of the constrained energy Hessian; $H_{ee} \in \mathbb{R}^{6\times 6}$. Here $H_{zz}^{-1} H_{ze}$ denotes a linear solve after the boundary and translation constraints have been applied. The relaxed energy satisfies $W^\star(\mathbf{e}) = \frac{1}{2}\mathbf{e}^{\mathsf{T}}\mathbf{C}\mathbf{e}$. This is the linearized energy-condensation form of periodic homogenization (Glaesener et al., 2020). Symmetric shear entries each carry one half of the engineering shear, consistent with the evaluator's conversion of its displacement-gradient tangent to a 6×6 Voigt matrix.

Pmmm symmetry eliminates normal–shear and off-diagonal shear couplings. We extract the nine remaining entries in the main-paper order

$$\mathbf{c}(\mathcal{T}) = (C_{11}, C_{22}, C_{33}, C_{12}, C_{13}, C_{23}, C_{44}, C_{55}, C_{66}). \tag{39}$$

With $E = 1$, the reported entries are effective stiffnesses normalized by the base material's Young's modulus.

Numerical evaluation. We solve periodic equilibrium with PETSc and an LU/MUMPS linear solver. Before the solve, we check finite coordinates, nonzero strut lengths, connectivity, and periodic pairing. Successful solves return the Voigt matrix and nine-component stiffness vector. Failed solves count as invalid candidates.

D METRIC DEFINITIONS AND COMPUTATION

We report dataset coverage, candidate validity, target success rate, and stiffness errors. Let $\mathcal{D}_{\mathrm{all}}$ be the full dataset of 1,492,505 canonical trusses. For method b, let $\mathcal{D}_b^{\mathrm{used}} \subseteq \mathcal{D}_{\mathrm{all}}$ contain the structures admitted by its native representation and adapter checks into its training or evaluation loader. These checks include the supported vertex-coordinate degrees of freedom and allowable edge-connectivity and topology complexity. Dataset coverage measures this operational representational support over canonical structures before generation; candidate validity separately measures whether sampled outputs pass structural and FEM checks. Dataset coverage is

$$\mathrm{DatasetCoverage}(b) = \frac{\left|\mathcal{D}_b^{\mathrm{used}}\right|}{|\mathcal{D}_{\mathrm{all}}|} \times 100\%. \tag{40}$$

For each target with a selected FEM-successful response, indexed by $n = 1, \ldots, N$, let $\mathbf{c}^{\star(n)}$ be the requested stiffness and $\widehat{\mathbf{c}}^{(n)}$ the selected FEM response. Their components $c_j^{\star(n)}$ and $\widehat{c}_j^{(n)}$, $j = 1, \ldots, 9$, follow the main-paper order. For evaluation, $R_j > 0$ is the frozen difference between the maximum and minimum of component j in the training split. The normalized component error is

$$d_{n,j} = \frac{\left|\widehat{c}_j^{(n)} - c_j^{\star(n)}\right|}{R_j}. \tag{41}$$

We compute the per-target errors as

$$\mathrm{NRMSE}_n = \sqrt{\frac{1}{9}\sum_{j=1}^{9} d_{n,j}^2}, \qquad \mathrm{NMAE}_n = \frac{1}{9}\sum_{j=1}^{9} d_{n,j}, \qquad e_{\infty,n} = \max_j d_{n,j}. \tag{42}$$

The reported NRMSE and NMAE are arithmetic means of these per-target values over the N targets with a selected FEM-successful response. For MAPE, exact-zero target components are masked before computing relative errors. Let $\mathcal{I} = \{(n, j) : c_j^{\star(n)} \neq 0,\ 1 \le n \le N,\ 1 \le j \le 9\}$ and $M = |\mathcal{I}| > 0$ be the number of unmasked target-component pairs. We compute

$$\mathrm{MAPE} = \frac{1}{M}\sum_{(n,j)\in\mathcal{I}} \frac{\left|\widehat{c}_j^{(n)} - c_j^{\star(n)}\right|}{\left|c_j^{\star(n)}\right|}. \tag{43}$$

For inverse-design evaluation, let $N_{\mathrm{gen}} > 0$ count all requested candidate slots, including generation failures, and N_{valid} count those yielding structures that pass all structural and FEM checks. Let $N_{\mathrm{req}} > 0$ count requested targets and N_{succ} count those with at least one FEM-successful candidate. The success measures are

$$\mathrm{CandidateValidity} = \frac{N_{\mathrm{valid}}}{N_{\mathrm{gen}}} \times 100\%, \qquad \mathrm{TargetSuccessRate} = \frac{N_{\mathrm{succ}}}{N_{\mathrm{req}}} \times 100\%. \tag{44}$$

Candidate validity uses the same binary indicator v_i^{val} as the SMCO validity reward. Failures remain in the success-rate denominators. For the property-error averages, $N = N_{\mathrm{succ}} > 0$; targets with no successful response have no stiffness error to aggregate.

Single-candidate and best-of-16 evaluations request $k = 1$ and $k = 16$ candidates per target, respectively. Best-of-k selects the candidate with the lowest NRMSE among those passing all structural and FEM checks, using the same candidate budget, frozen scales, and selection rule for the compared models. Timing covers the native target-to-output generation procedure and excludes model loading and FEM evaluation.

E DETAILED EXPERIMENTAL SETTINGS

E.1 EXTERNAL BASELINE TRAINING AND ADAPTATION

We retrain UniMate, UniTruss, GraphMetaMat, and TrussGPT on the current dataset following their respective official training recipes (Zhan et al., 2025; Zheng et al., 2023a; Maurizi et al., 2025; Lin

et al., 2026). All baselines use stiffness labels recomputed with ae108 at $E = 1$, $\nu = 0.3$, and relative density $\rho = 0.15$, under the physical protocol in Section C. Training uses the subset of the training split admitted by each method's representation and adapter; Table 7 gives the resulting counts and adaptation rules. Checkpoints are selected on the current dataset's validation split using each baseline's own validation metric. Evaluation requests the target stiffness vectors of all 69,262 structures in the test split.

Stiffness conditioning. All four baselines use the main-paper component order for the nine stiffness inputs. A common conditioning architecture standardizes each component as $(c_j - \mu_j)/\sigma_j$ using frozen training-set means and standard deviations, and produces nine 1,024-dimensional embeddings. Each baseline trains its own conditioner weights and projects these embeddings into its native interface, as detailed in Table 7. UniTruss additionally reorders the target to $(C_{11}, C_{12}, C_{13}, C_{22}, C_{23}, C_{33}, C_{44}, C_{55}, C_{66})$ for its native property predictor and latent-search objective. Input standardization is separate from the component-wise max–min normalization used for evaluation errors in Section D.

Table 7: Method-specific baseline adaptation and generation settings. Training counts are measured after admission within the 1,336,666-structure training split. Search budgets describe each native target-to-output procedure.

Method	Training data and structural adapter	C9 interface	Generation or search budget
UniMate	39,191 structures. Accept FEM-eligible complete unit-cell graphs with fewer than 40 nodes. Generate continuous coordinates in $[0, 1]^3$ and edge connectivity.	Project each of the nine embeddings to a 16-dimensional property token for the shared codebook.	10 generation stages with 100 Langevin steps per stage.
UniTruss	856,932 structures from the UniTruss-source subset. Decode the native 27-node adjacency and coordinate offsets; retain active nodes and expand the octant to the full cell.	Mean-pool the nine embeddings and project to a 48-dimensional latent initializer; retain the native property predictor for search.	100 latent initializations, each optimized for 1,200 steps.
GraphMetaMat	123,388 structures. Admit connected tetrahedral graphs with 3–4 nodes that can be mapped to the native action space with coordinate displacement at most 0.05 in native coordinates and at most one edge deletion; tessellate and relabel the adapted structures.	Mean-pool the nine embeddings and project to the native 256-dimensional policy condition.	128 Monte Carlo tree-search runs with exploration coefficient $c_{\mathrm{puct}} = 2.5$.
TrussGPT	479,920 structures. Require compatibility with the fixed 15-node octant and an exact fold–expand round trip. Decode adjacency through the native tokenizer and expand to the full cell.	Project the nine embeddings to 768-dimensional prefix tokens for the autoregressive designer.	Autoregressive sampling with temperature 1.2 and top-$p = 0.9$; additional best-of-k sampling is also evaluated.

Output checks and candidate selection. TrussGPT rejects out-of-vocabulary token sequences and empty decoded edge sets; its adjacency threshold is 0.6. UniTruss uses an adjacency threshold of 0.5 and rejects empty decoded structures. GraphMetaMat maps native coordinates from $[-1, 1]^3$ to $[0, 1]^3$ and rejects empty graphs or malformed coordinates. These three adapters project expanded coordinates to the common integer grid and check unique nodes, legal nonduplicate edges, symmetry consistency, physical connectivity, minimum node degree of two, and a connected periodic quotient with minimum degree two and winding rank three. UniMate retains continuous coordinates, rejects malformed, nonfinite, or out-of-cell coordinates and empty edge sets, and checks physical connectivity and minimum degree two before FEM. A candidate is valid only after successful FEM evaluation.

Native search ranks its internal proposals using the method's own objective; when multiple output candidates are evaluated by FEM for a target, the final response is the valid candidate with the lowest NRMSE. Failed requests remain in the validity denominator.

E.2 TrussOrbit model and optimization settings

Table 8 reports the model, training schedule, SMCO hyperparameters, and rollout sampling settings. Table 9 gives the configurations used for the policy-objective comparisons. Training group sizes and optimizer settings vary by policy objective.

Table 8: Detailed model, training, and sampling settings.

Stage	Parameter	Setting
Model	Transformer	16 blocks; 16 heads; hidden size 1,024; context 1,536 tokens
Pretraining	Steps; learning rate; warm-up; weight decay	200,000; 10^{-4}; 6,000 steps; 0.01
SFT	Steps; learning rate; schedule	200,000; 3×10^{-5}; conditioner-only 20,000, then joint 180,000 steps
SMCO	Steps; learning rate; initialization	2,000; 2×10^{-6}; conditional SFT checkpoint
SMCO	Batch; candidates k	32; 32 per target
SMCO	τ_u	0.05
SMCO	τ_b	0.05
SMCO	λ_{val}	0.5
SMCO	Clip $(\epsilon_{\text{l}}, \epsilon_{\text{h}})$	$(0.001, 0.001)$
SMCO	Advantage ε_A	10^{-12}
Rollout	Temperature; top-p; top-k	1.0; 1.0; -1
Rollout	Training candidate budget	$k = 32$
Evaluation	Sampling budget	$k = 1$ and $k = 16$
Evaluation	Token limits	1,536 context; 1,024 codec limit
Compute	Hardware; software	4 NVIDIA A100 80 GB GPUs; veRL/HybridFlow, SGLang, FlashAttention-2
Compute	Wall time	Approx. 6 / 20 / 60 h for pretraining / SFT / SMCO

Table 9: Policy-optimization settings used in the experiments. Baseline update rules follow the cited methods. The direct-reward control inherits the SMCO settings and changes its advantage formula.

Method	Prompt batch; group size	Learning rate; duration	KL; clipping	Sampling and method-specific setting
RLOO (Ahmadian et al., 2024)	32 prompts ; $k = 2$	10^{-6}; 2,000 steps	$\beta = 0.05$; unclipped	Temperature 1.0; top-$p = 1.0$; leave-one-out sequence advantage
GRPO (Shao et al., 2024)	32 prompts; $k = 5$; mini-batch 256	10^{-6}; 2,000 steps	$\beta = 0.001$ (low-variance KL loss); clip 0.2	Temperature 1.0; group-normalized scalar reward
GSPO (Zheng et al., 2025)	32 prompts; $k = 16$; mini-batch 128	10^{-6}; 2,000 steps	No KL loss; clip $(3 \times 10^{-4}, 4 \times 10^{-4})$	Temperature 1.0; sequence-level ratio
SetPO (Li et al., 2026a)	32 prompts; $k = 6$; mini-batch 32	3×10^{-6}; 2,000 steps	$\beta = 10^{-4}$; clip 0.2	Temperature 0.9; $\lambda_{\text{SetPO}} = 0.05$
SMCO	32 prompts; $k = 32$	2×10^{-6}; 2,000 steps	No KL loss; clip $(0.001, 0.001)$	Temperature 1.0; top-$p = 1.0$; $\tau_u = \tau_b = 0.05$, $\lambda_{\text{val}} = 0.5$
Direct-reward control	32 prompts; $k = 32$	2×10^{-6}; 2,000 steps	Same as SMCO	Same as SMCO; use $\mathbf{A}^{\text{direct}}$ in Section F.6

Checkpoint selection evaluates generated candidates on the validation split with the common validity and stiffness-matching checks. For each method, we retain the checkpoint with the best validation target-level Hit@0.05; ties are resolved by lower validation NRMSE among successful responses.

The selected checkpoint is then evaluated on the test split using the metrics and aggregation rules in Section D. Policy-objective methods share the test targets, $k = 1$ evaluation budget, decoding settings, validity checks, and FEM evaluator; their training batch and rollout group settings are given in Table 9.

F Additional Experiments

F.1 Multiple-Solution Generation

We evaluate how many topologically distinct structures match each stiffness target within a fixed generation budget. The protocol uses the frozen set of 69,262 test targets, the training-set normalization ranges in Section D, and the common structural checks and ae108 settings. Table 10 reports the results for SFT, SFT+GSPO, SFT+SetPO, and TrussOrbit (SFT+SMCO) at $k = 16$.

F.1.1 Physically matched topology counts

For target n, use the SMCO notation k for the candidate budget, and let $g_{n,i} = 1$ indicate that candidate i passes all structural and FEM checks. A matched candidate has $g_{n,i} = 1$ and $\mathrm{NRMSE}_{n,i} \le \tau$, where the discrepancy compares its FEM stiffness with the requested target using Section D. We use $\tau = 0.05$ as the primary threshold and $\tau = 0.10$ for sensitivity analysis. These thresholds use min–max normalization. Define the matched structures and their number of distinct topologies as

$$\begin{aligned} \mathcal{H}_n^{(k)}(\tau) &= \{\mathcal{T}_{n,i} : 1 \le i \le k,\ g_{n,i} = 1,\ \mathrm{NRMSE}_{n,i} \le \tau\}, \\ H_n^{(k)}(\tau) &= \left| \mathcal{H}_n^{(k)}(\tau)/\cong \right| . \end{aligned} \tag{45}$$

Here $\cong$ denotes isomorphism of the expanded full-cell undirected graphs after coincident nodes are merged. Four-round WL signatures computed from unlabelled connectivity provide a prefilter (Shervashidze et al., 2011); exact graph-isomorphism checks within each signature group determine the final classes. Each class groups node relabelings and geometric variants with isomorphic connectivity. This test is applied to all matched candidates in the requested group. It counts distinct matched topologies across the group, complementing the stiffness error of the single response selected in best-of-k evaluation.

With $N_{\mathrm{req}} = 69{,}262$, the mean number of distinct matched topologies and the fraction of targets with at least m such topologies are

$$\begin{aligned} \overline{H}^{(k)}(\tau) &= \frac{1}{N_{\mathrm{req}}} \sum_{n=1}^{N_{\mathrm{req}}} H_n^{(k)}(\tau), \\ \mathrm{Hit}_{\ge m}^{(k)}(\tau) &= \frac{100\%}{N_{\mathrm{req}}} \sum_{n=1}^{N_{\mathrm{req}}} \mathbf{1}\{H_n^{(k)}(\tau) \ge m\}, \qquad m \in \{1, 2, 4\}. \end{aligned} \tag{46}$$

Targets without a match contribute zero to both measures. Failed and repeated outputs retain their slots and consume the budget. For the policy comparison, all four models use the same targets, $k = 16$, decoding settings, and physical evaluator. The two threshold panels use the same candidate records.

At the primary threshold, TrussOrbit yields 39.4% more distinct matched topologies per target than SFT at the fixed candidate budget. Its advantage persists under the relaxed threshold and extends to targets with several matched alternatives. The tied single-hit rate under the relaxed threshold, together with the higher matched-topology count, shows that TrussOrbit broadens the choice of feasible alternatives even when both methods already find a match.

F.1.2 Candidate records for reproducibility

The generation evaluation requires the target identifier and stiffness vector, candidate index in its original generation order, canonical sequence or native structure, expanded node coordinates and connectivity, structural-check outcomes, FEM status and nine-component response, and topology

Table 10: Distinct topology hits at $k = 16$. $\mathrm{Hit}_{\geq m}$ counts targets with at least m distinct topologies satisfying the stiffness threshold. All target-level statistics include the complete requested test set. Bold indicates the best value in each metric column within each threshold panel; all ties are highlighted.

Training strategy	$\overline{H}$	$\mathrm{Hit}_{\geq 1}$ (%)	$\mathrm{Hit}_{\geq 2}$ (%)	$\mathrm{Hit}_{\geq 4}$ (%)
(a) Primary threshold: $\tau = 0.05$				
SFT	6.1135	97.58	91.15	71.41
SFT+GSPO	5.9015	97.35	90.52	68.72
SFT+SetPO	5.8729	97.28	88.41	65.27
SFT+SMCO (Ours)	**8.5249**	**98.68**	**94.30**	**85.43**
(b) Sensitivity threshold: $\tau = 0.10$				
SFT	9.9014	**99.99**	99.32	94.06
SFT+GSPO	9.8421	99.23	98.56	93.77
SFT+SetPO	9.8753	99.02	98.35	93.12
SFT+SMCO (Ours)	**12.7586**	**99.99**	**99.84**	**96.27**

class. Bind each evaluation to the checkpoint, decoding configuration, random seeds, frozen normalization ranges, and requested target list. Retain every requested slot. Full candidate groups supply the topology-hit counts in Equations 45 and 46.

F.2 Single-Candidate Target Hit Rates for External Baselines

Table 11 uses $\mathrm{Hit}^{(k)}_{\geq 1}(\tau)$ from Equation 46 to compare external baselines with TrussOrbit at $k = 1$. Each method receives the same 69,262 test targets and produces one output candidate per target using the native generation or search procedure in Section E.1. A target is a hit when its candidate passes all structural and FEM checks and has NRMSE at most τ, using the frozen training-set component-wise max–min ranges. Both threshold rates use all requested targets as the denominator; generation failures, invalid structures, FEM failures, and valid outputs above the threshold count as misses. At $k = 1$, no candidate selection or topology deduplication is required.

Table 11: Single-candidate target hit rates over all $N_{\mathrm{req}} = 69{,}262$ test targets. Both hit rates are percentages, and higher is better. Bold indicates the best value in each hit-rate column.

Method	k	$\mathrm{Hit}_{\geq 1}$ @0.05 (%) ↑	$\mathrm{Hit}_{\geq 1}$ @0.10 (%) ↑
UniMate	1	0.82	13.89
UniTruss	1	7.42	27.62
GraphMetaMat	1	10.99	34.41
TrussGPT	1	0.41	1.02
TrussOrbit (Ours)	1	**54.88**	**87.30**

At the primary threshold, TrussOrbit reaches a target hit rate of 54.88%, compared with 10.99% for the strongest external baseline, GraphMetaMat (Table 11). Its lead persists under the relaxed threshold. Including every requested target makes this comparison sensitive to both generation failures and insufficient stiffness matching, complementing the error averages over successful responses in the main paper.

F.3 Crossed Ablation of Representation and Training

Table 12 crosses SFT and SFT+SMCO with three edge representations to examine how the effect of policy optimization depends on the representation. VV directly outputs the coordinates of both strut endpoints; UV outputs node-index pairs (U, V); EOTR encodes edge orbits. Each representation has its own pretrained base model and corresponding SFT checkpoint; SMCO continues training from that checkpoint. The comparison shares the training split, backbone architecture, stiffness-conditioning design, test targets, and evaluation protocol, with metrics and settings in Sections D and E. Evaluation

uses one candidate per target ($k = 1$), includes failed slots in candidate validity, and averages stiffness errors over FEM-successful responses.

Table 12: Crossed ablation of training strategy and edge representation. EOTR with SFT+SMCO is TrussOrbit (Ours). EOTR results reproduce the main-paper policy ablation. Bold indicates the best reported value in each metric column; all ties are highlighted.

Training strategy	Edge representation	Validity (%) ↑	NMAE ↓	NRMSE ↓	MAPE ↓
SFT	VV	87.45	0.2055	0.2241	0.3895
SFT	UV	89.75	0.0824	0.1239	0.3306
SFT	EOTR	92.64	0.0371	0.0479	0.2736
SFT+SMCO	VV	88.31	0.1608	0.1976	0.3617
SFT+SMCO	UV	90.62	0.0777	0.0891	0.2921
SFT+SMCO (Ours)	EOTR	**93.27**	**0.0354**	**0.0432**	**0.2623**

With SFT, EOTR reaches 0.0479 NRMSE, the lowest among the three representations. It also has higher validity than the other representations. Adding SMCO improves validity and all three stiffness errors within each representation, while EOTR remains the strongest under both training strategies.

F.4 Vertex Quantization Resolution

We evaluate the stiffness error introduced by vertex-coordinate quantization in EOTR at three grid resolutions. For each structure, we retain the original continuous coordinates from the pre-conversion metadata as the reference geometry and construct its quantized counterpart. Both geometries are evaluated with ae108 under the same physical settings in Section C. The FEM stiffness of the continuous geometry serves as ground truth; the quantized geometry's FEM stiffness is compared against it using the NRMSE and frozen training-set component-wise max–min ranges in Section D. A grid with Q coordinate levels on $[0, 1]$ has step size $1/(Q - 1)$.

Table 13: FEM stiffness error introduced by EOTR vertex quantization relative to the original continuous-coordinate geometry. Entries are the median and 95th percentile of per-structure NRMSE, expressed as percentages and rounded to two decimal places. Bold indicates the lowest error in each metric column.

Coordinate grid	Step size	Median NRMSE (%) ↓	95th-percentile NRMSE (%) ↓
Q257	$1/256$	0.28	0.95
Q513	$1/512$	0.15	0.48
Q1025	$1/1024$	**0.08**	**0.23**

At EOTR's selected coordinate resolution, quantization yields a median NRMSE of 0.08% and a 95th-percentile NRMSE of 0.23% (Table 13). Refining the grid reduces both typical and upper-tail stiffness errors. The selected resolution therefore preserves the continuous geometry's FEM response more closely than the coarser grids under the common physical evaluator.

F.5 Best-of-k Ablation of SFT and TrussOrbit

Table 14 compares SFT and TrussOrbit at candidate budgets $k \in \{1, 2, 4, 8, 16, 32\}$ using the same test targets, decoding settings, and physical evaluator. For each target, best-of-k selects the candidate with the lowest NRMSE among those passing all structural and FEM checks. Target success is the percentage of requested targets with at least one valid candidate, as defined in Section D; at $k = 1$, it equals candidate validity. NRMSE is averaged over targets with a valid selected response and uses the frozen training-set component-wise max–min ranges. Time measures generation of the full k-candidate group per requested target, excluding model loading and FEM evaluation, under the timing convention in Section D.

At $k = 16$, TrussOrbit reduces best-of-k NRMSE by 15.0% relative to SFT. Its accuracy advantage holds across every tested candidate budget. Both methods reach full target success as the budget increases; at the largest budget, TrussOrbit retains lower error with slightly less generation time.

Table 14: Best-of-k performance of SFT and TrussOrbit. Time covers generation of all k candidates per target. Bold marks the better value between methods for each metric at the same k; all ties are highlighted.

	SFT			SFT+SMCO (Ours)		
k	Time (ms/target) ↓	Target success (%) ↑	NRMSE ↓	Time (ms/target) ↓	Target success (%) ↑	NRMSE ↓
1	**19.60**	92.64	0.0479	20.70	**93.27**	**0.0432**
2	65.83	99.25	0.0377	**64.25**	**99.36**	**0.0346**
4	**215.60**	**99.99**	0.0302	235.19	**99.99**	**0.0291**
8	**525.31**	**100.00**	0.0254	561.02	**100.00**	**0.0247**
16	814.57	**100.00**	0.0220	**796.71**	**100.00**	**0.0187**
32	1489.42	**100.00**	0.0195	**1429.24**	**100.00**	**0.0172**

F.6 Direct-Reward Control for SMCO

We compare TrussOrbit with a direct-reward control initialized from the same SFT checkpoint to isolate null-slot marginal credit. Both use the SMCO configuration in Table 9 and the same target sampling, physical checks, and validation checkpoint-selection rule. The control retains GSPO, the two-stage group normalization, and both reward channels, replacing marginal contributions with each candidate's own rewards:

$$\mathbf{A}^{\mathrm{direct}} = \mathcal{N}\big(\mathcal{N}(\mathbf{q}) + \lambda_{\mathrm{val}}\mathcal{N}(\mathbf{v}^{\mathrm{val}})\big)\,. \tag{47}$$

Here $\mathcal{N}(\mathbf{z})$ collects the components $\mathcal{N}_i(\mathbf{z})$, and $\mathbf{v}^{\mathrm{val}}$ collects the binary validity indicators. The direct reward uses v_i^{val}, whereas SMCO's validity margin is v_i^{val}/k before normalization. Table 15 reports single-candidate and best-of-16 results, with topology thresholds from Section F.1.

Table 15: SMCO marginal-credit ablation against a direct-reward control; both continue from the same SFT checkpoint. $\overline{H}$ is the mean number of distinct matched topologies per target. Bold indicates the best value in each metric column within each panel; all ties are highlighted.

(a) Single-candidate results ($k = 1$)

Training strategy	Candidate validity (%) ↑	NMAE ↓	NRMSE ↓	MAPE ↓
SFT+Direct reward	92.87	0.0391	0.0471	0.2839
SFT+SMCO (Ours)	**93.27**	**0.0354**	**0.0432**	**0.2623**

(b) Best-of-16 results, primary threshold ($\tau = 0.05$)

Training strategy	Target success (%) ↑	$\mathrm{Hit}_{\geq 1}$ (%) ↑	$\overline{H}$ ↑	$\mathrm{Hit}_{\geq 4}$ (%) ↑	NRMSE ↓
SFT+Direct reward	**100.00**	93.02	7.8454	83.21	0.0354
SFT+SMCO (Ours)	**100.00**	**98.68**	**8.5249**	**85.43**	**0.0187**

(c) Best-of-16 topology sensitivity ($\tau = 0.10$)

Training strategy	$\mathrm{Hit}_{\geq 1}$ (%) ↑	$\overline{H}$ ↑	$\mathrm{Hit}_{\geq 4}$ (%) ↑
SFT+Direct reward	**99.99**	10.3543	92.43
SFT+SMCO (Ours)	**99.99**	**12.7586**	**96.27**

With one candidate per target, TrussOrbit lowers NRMSE from 0.0471 to 0.0432 relative to the direct-reward control. It also improves candidate validity and the other stiffness-error measures. Under matched training settings, these gains support the contribution of marginal credit beyond the shared rewards and policy-update rule.

With best-of-16 selection, TrussOrbit lowers NRMSE from 0.0354 to 0.0187 relative to the direct-reward control. Both methods produce a valid response for every target, while TrussOrbit more often meets the primary stiffness threshold and provides more distinct matched alternatives. Its matched-topology advantage persists at the relaxed threshold even when the single-hit rates tie.

F.7 Additional Qualitative Comparisons

Across the selected targets in Figure 10, the displayed TrussOrbit candidates span 0.05–3.18% NRMSE. They exhibit different internal connectivity while closely matching the same target stiffness. The displayed baseline outputs have larger stiffness deviations or fail the validity checks.

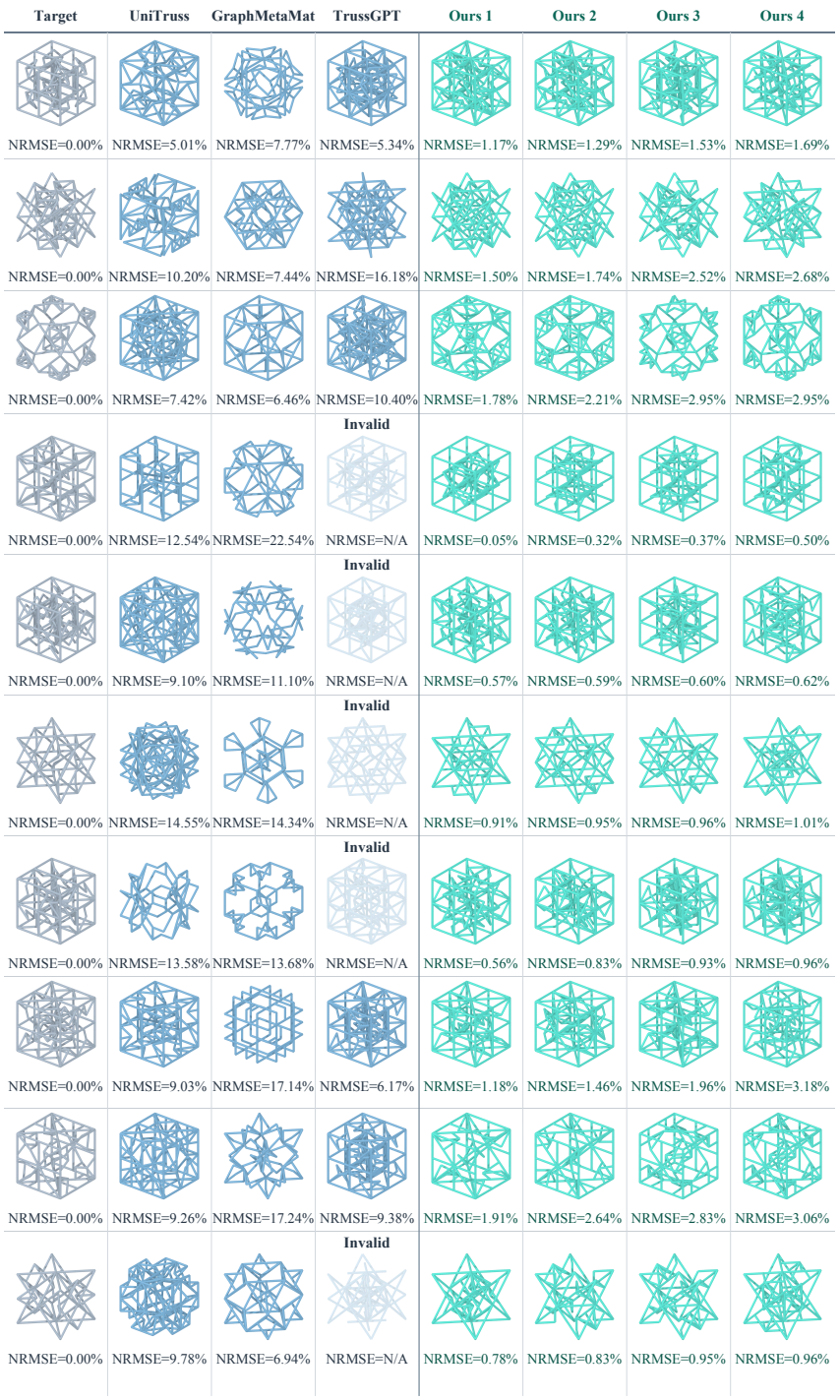


Figure 10: Additional inverse-design comparisons on ten target stiffnesses. Ours 1–4 show four TrussOrbit candidates per target. Labels report NRMSE; N/A denotes outputs marked invalid.

F.8 Failure Cases

Figure 11 illustrates failure cases produced by TrussOrbit (SFT+SMCO) during inverse reconstruction. Some outputs pass the structural checks but deviate from the requested stiffness, showing that validity alone does not ensure an accurate response. Other outputs have disconnected topology and fail the structural checks required for FEM evaluation.

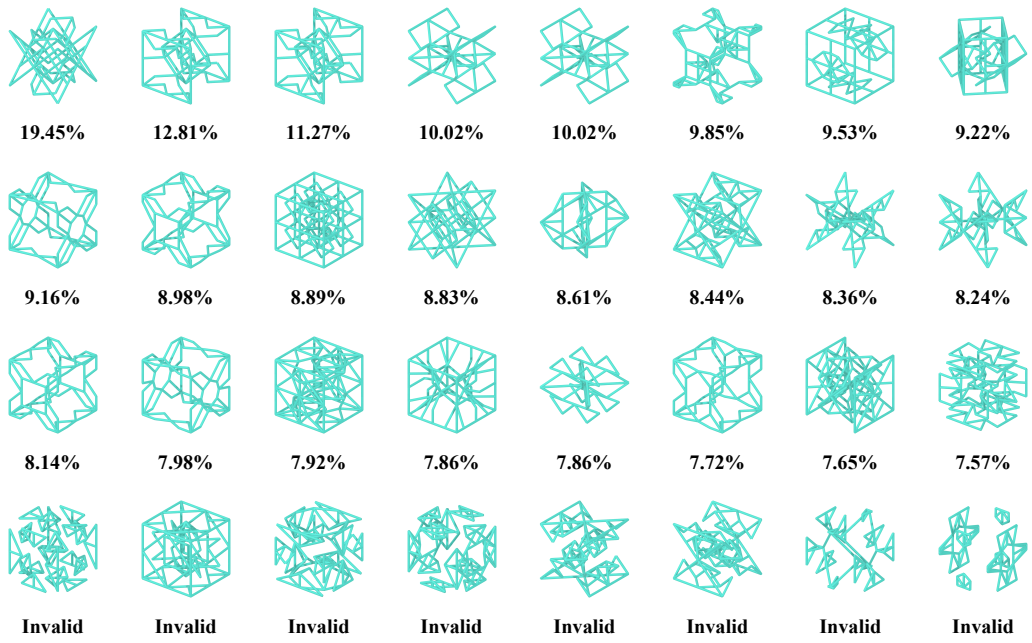


Figure 11: Failure cases produced during inverse reconstruction with TrussOrbit (SFT+SMCO). The upper rows show structurally valid candidates with stiffness mismatches; labels report NRMSE. The bottom row shows disconnected candidates that fail structural checks.