

METASYMBO: Multi-Agent Language-Guided Metamaterial Discovery via Symbolic Latent Evolution

Jianpeng Chen¹, Wangzhi Zhan¹, Dongqi Fu², Junkai Zhang³, Zian Jia^{4,5}, Ling Li⁵
Wei Wang³, Dawei Zhou¹

¹Virginia Tech ²Meta AI ³University of California Los Angeles

⁴Princeton University ⁵University of Pennsylvania

jianpengc@vt.edu

Abstract

Metamaterial discovery seeks microstructured materials whose geometry induces targeted mechanical behavior. Existing inverse-design methods can efficiently generate candidates, but they typically require explicit numerical property targets and are less suitable for early-stage exploration, where researchers often begin with incomplete constraints and qualitative intents expressed in natural language. Large language models can interpret such intents, but they lack geometric awareness and physical property validity. To address this gap, we propose **METASYMBO**, a multi-agent framework for language-guided **Metamaterial** discovery via **Symbolic**-driven latent ev**Olution**. Specifically, METASYMBO contains three agents: a Designer that interprets free-form design intents and retrieves a semantically consistent *scaffold*, a Generator that synthesizes candidate microstructures in a disentangled latent space, and a Supervisor that provides fast property-aware feedback for iterative refinement. To move beyond the limitations of reproducing known samples from literature and training data, we further introduce symbolic-driven latent evolution, which applies programmable operators over disentangled latent factors to compose, modify, and refine structures at inference time. Extensive experiments demonstrate that (i) METASYMBO improves structural validity by up to 34% in symmetry and nearly 98% in periodicity compared to state-of-the-art baselines; (ii) METASYMBO achieves about 6–7% higher language-guidance scores while maintaining superior structure novelty compared to advanced reasoning LLMs; (iii) qualitative analyses confirm the effectiveness of symbolic logic operators in enabling programmable semantic alignment; and (iv) real-world case studies on auxetic, high-stiffness metamaterial design further validate its practical capability.

1 Introduction

Metamaterials, an emerging microstructured category of materials, are receiving increasing attention due to their capability to achieve extraordinary mechanical properties, exhibiting wide applications in various fields, such as biomedical devices, transportation systems, robotics, *etc.* (Paul, 2010; Engheta & Ziolkowski, 2006; Jia et al., 2020). A central goal of metamaterial discovery is to identify microstructures with desired mechanical responses, such as target elastic moduli or Poisson’s ratios (Deng et al., 2022). As the design space of possible microstructures is extremely large, recent research has increasingly shifted toward data-driven methods to improve the efficiency of metamaterial discovery. In particular, variational autoencoders (VAEs) (Kingma & Welling, 2014), diffusion models (DMs) (Podell et al., 2023; Fu et al., 2024; Zhan et al., 2025), and generative adversarial networks (GANs) (Tian et al., 2022; Pahlavani et al., 2024) have been applied to learn structure–property relationships for inverse metamaterial design tasks (Zheng et al., 2023; Xie et al., 2022; Ma et al., 2019). However, although these methods are effective for exploring large design spaces, they usually require explicit numerical property specifications and cannot naturally interpret

free-form language descriptions or implicit domain knowledge from the literature, which are often critical in the early stage of metamaterial design.

Recent studies (Ha et al., 2023; Chen et al., 2024b) indicate that early-stage metamaterial discovery often begins before precise numerical targets are available, with incomplete information, evolving constraints, and qualitative functional goals. An engineer may initially seek a structure that is “lightweight, impact-absorbing, and recoverable under compression,” while suitable ranges of elastic modulus, Poisson’s ratio, density, anisotropy, and manufacturability may only become clear after candidate structures are proposed and evaluated. Traditional inverse-design methods that require explicit numerical property inputs are therefore less suitable for such exploratory stages and may even fail to produce valid designs under poorly specified conditions (Jin et al., 2023; Ronellenfitsch et al., 2019); moreover, fixing numerical targets too early can unnecessarily restrict exploration to a narrow realization of a broader functional intent. Natural language instead provides a flexible interface for expressing qualitative requirements and trade-offs without specifying every mechanical quantity in advance, while LLMs offer additional capabilities for interpreting such intents and retrieving relevant domain knowledge (Tian et al., 2025; Narayanan et al., 2025), as demonstrated by models such as GPT-4o (Hurst et al., 2024) and DeepSeek (Guo et al., 2025). Our goal is therefore not to replace numerical inverse design when precise targets are already known, but to address the preceding exploratory regime in which application-level intent must first be translated into plausible structural hypotheses and progressively refined using geometric and property feedback; nevertheless, LLMs alone remain insufficient because they lack explicit geometric awareness and cannot reliably enforce physical validity.

These observations reveal a persistent modality gap: LLM agents are strong at interpreting language-level design intents, whereas geometry-aware generators are effective at producing structurally valid and physically realistic designs, but neither alone can bridge these domains. Therefore, a natural research question arises, *can qualitative language-level design intent be systematically translated into valid metamaterial geometries through coordinated language, geometry, and property reasoning?*

To achieve this, we identify two critical challenges. **C1:** *How can we bridge the modality gap, considering the large discrepancy among modalities?* Language-guided metamaterial discovery involves three distinct modalities: language, geometry, and properties. For language modality, interpreting qualitative design intents, such as “strong but flexible”, requires robust reasoning capabilities and substantial domain expertise. For geometric modality, designing microstructures demands an understanding of structural consistency and adherence to physical constraints. For property modality, creating materials to achieve targeted properties requires comprehensive knowledge of the mechanical properties. **C2:** *How can the system explore candidate geometries beyond direct retrieval or reproduction of training examples?* Both LLMs and learned generators are biased toward designs represented in their literature or training distributions. A useful exploration system should therefore recombine known structural motifs while retaining geometric validity, enabling broader hypothesis generation rather than simple memorization.

To address the two challenges, we propose **METASYMBO**, a multi-agent collaborative framework utilizing symbolic-driven latent evolution for metamaterial discovery. Specifically, for **C1**, we introduce three specialized agents, each focusing on one modality. Agent Designer, powered by an LLM, provides language reasoning to interpret prompts and query the domain literature. Agent Generator is a generative model specialized in geometric modality, enabling exploration within vast valid geometric design spaces. Agent Supervisor integrates property prediction and language reasoning, facilitating property-driven design. Moreover, a collaboration mechanism is developed for joint structure optimization. For **C2**, motivated by the programmable methods (Bastek et al., 2022; Zhao et al., 2025), we propose a symbolic-driven latent evolution approach to optimize geometries toward language semantics in Generator, achieving effective hypothesis exploration outside the literature domain and the training data domain. The contributions can be summarized:

- **Conceptual Contribution:** We unify language guidance, geometric generation, and property-aware reasoning in a single pipeline to enable the practical metamaterial discovery workflows starting with a conceptual-level idea.

- **Methodology Contribution:** We propose METASYMBO, a multi-agent framework for efficient and robust metamaterial discovery, together with a symbolic-driven latent evolution module for inference-time cross-domain exploration.
- **Benchmark Contribution:** We benchmark METASYMBO against state-of-the-art LLMs and generative models, showing superior structural validity, design diversity, and language-guidance effectiveness. These quantitative comparisons validate the effectiveness.
- **Practical Contribution:** Case studies demonstrate the end-to-end applicability of METASYMBO in realistic metamaterial design workflows, from prompt refinement and structure generation to downstream physical validation of 3D printed samples.

2 Preliminary

Metamaterial Design. Metamaterials, as shown in Figure 1, are typically modeled as lattice structures, *i.e.*, periodic arrangements of unit cells composed of trusses in geometric space (Chen et al., 2025b). Formally, a metamaterial can be represented as $M = (\mathbf{L}, \mathcal{U})$, each unit cell $\mathcal{U} = (\mathbf{P}, E)$ consists of node coordinates $\mathbf{P} \in \mathbb{R}^{N \times 3}$ and edge set E specifying strut connections, and the lattice vectors $\mathbf{L} = [l_1, l_2, l_3]^T \in \mathbb{R}^{3 \times 3}$ define the periodic tiling of unit cells in 3D space. The corresponding mechanical properties are denoted by $\mathbf{y} \in \mathbb{R}^{d_y}$. The central goal of metamaterial discovery is to design structures with tailored properties that satisfy specific functional requirements.

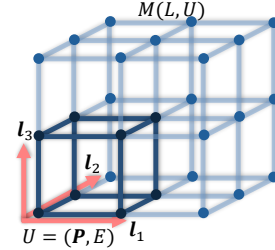


Figure 1: Truss lattice.

Limitations of Existing Works. Recent metamaterial discovery has increasingly relied on data-driven methods that learn structure–property relationships from data. Generative models such as VAEs and DMs typically optimize a conditional autoencoder $q_\phi(\mathbf{z} \mid M)$ and $p_\theta(M \mid \mathbf{z}, \mathbf{y})$ under a Gaussian prior $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. While effective for inverse design, these methods usually depend on explicit numerical property conditions (*i.e.*, $\mathbf{y}$) and often entangle geometric and semantic information in a shared latent space, limiting controllability and generation quality.

LLM-based approaches are trained on large volumes of domain literature and therefore offer strong potential for retrieving candidates from broad design spaces (Hayes et al., 2024; Gruver et al., 2024; Li et al., 2025b;a; Zholus et al., 2025; Narayanan et al., 2025). However, existing methods either adapt LLMs to strictly formatted material representations, which weakens their ability to understand free-form language, or simply prompt them to generate materials directly, causing them to favor reproducing known materials from the literature rather than proposing novel designs.

Problem Statement. Given a free-form language prompt, the goal of this work is to perform language-guided metamaterial design to generate structures whose mechanical properties satisfy the intent expressed in the prompt.

3 METASYMBO: Language-Guided Metamaterial Discovery

In this section, we present METASYMBO, a multi-agent framework for language-guided metamaterial discovery, as shown in Figure 2. To bridge the modality gap across language, geometry, and properties (Challenge C1), METASYMBO coordinates three specialized agents. **Designer** interprets the input prompt and retrieves a simple existing structure (called as *scaffold*), which serves as a semantic anchor between free-form language and geometric generation. Guided by this scaffold, **Generator** searches the design space in the proposed disentangled latent space, where semantic and geometric factors are explicitly separated for controllable and valid synthesis. To further enable exploration beyond known designs (Challenge C2), we introduce *symbolic-driven latent evolution*, which refines latent representations to align with scaffold semantics while expanding the reachable design space. Finally, **Supervisor** provides fast property prediction and language-based evaluation to assess generated candidates for iterative feedback, forming a closed-loop optimization process for language-guided metamaterial design.

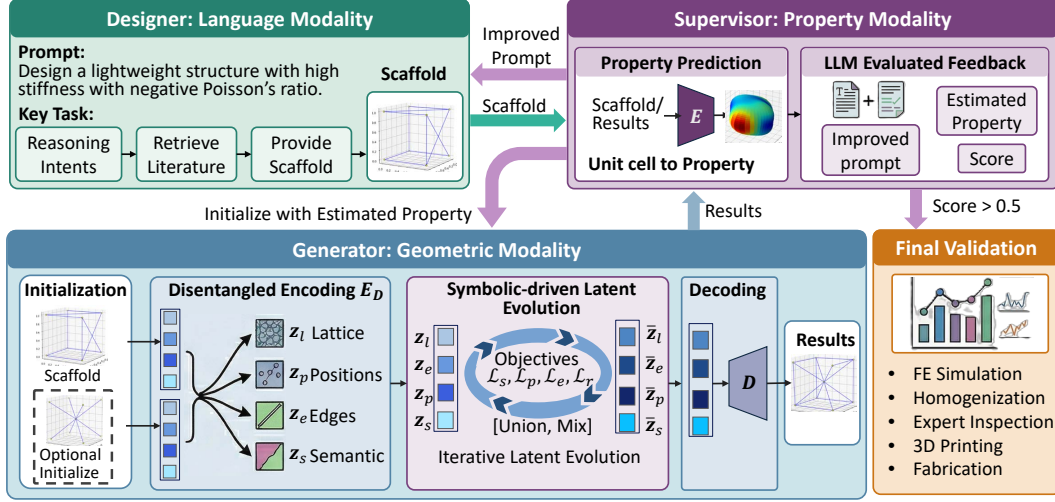


Figure 2: **Overview of METASYMBO.** Agent Designer translates the prompt into a scaffold, Agent Generator refines the design in latent geometric space via symbolic-driven latent evolution, and Agent Supervisor evaluates properties to provide fast iterative feedback.

3.1 Designer: Language-Modality Designer

To address the modality gap between conceptual-level language and complex geometric space, there are two central issues: (1) the initial metamaterial design idea might be vague and incomplete (Ha et al., 2023; Chen et al., 2024b), which requires the Designer to be equipped with a large amount of domain literature to infer the intent from the initial prompt; (2) after the Designer could successfully understand the prompt, the critical issue is how the model can translate the intent of prompt into geometric space.

Fortunately, LLMs have been demonstrated to be effective for understanding, retrieving, and reasoning in language space for domain-specific scientific discovery. These previous works demonstrate the capability of LLMs to infer the intent from a prompt: for example, the phrase “a hard material” statistically co-occurs with “high Young’s modulus” in the literature, allowing the model to map the qualitative adjective “hard” to a quantitative stiffness concept (Jin et al., 2023) (as examples validated in Appendix C.2). As for the second issue, although LLMs can render existing classic lattices, they often fail when it comes to more complex generation tasks and larger geometric design spaces. The experiments for LLMs in Table 1 show that LLMs tend to generate repeated structures, indicating their capability in successfully retrieving existing structures and limitations to explore large geometric design space. To balance this, we propose “scaffold”.

Scaffold, the output of Designer, refers to existing simple geometric structures that imply the core intent of the initial prompt. The high-level idea of it is to utilize the strong retrieval and language-understanding power of LLMs. For example, given a prompt for generating “stable” structure, we expect LLMs to retrieve a scaffold of an octet or triangle-like structure (as shown in Appendix C.2), rather than a cubic that is less stable. The model thus outputs a concise scaffold description that is semantically consistent with the prompt.

3.2 Generator: Geometry-Modality Synthesizer

Generator is the geometry-specialized component of METASYMBO. It receives the scaffold selected by Designer, together with optional property guidance from Supervisor, and synthesizes a new metamaterial geometry. Unlike Designer and the LLM evaluator in Supervisor, Generator does not perform language reasoning; its latent variables and symbolic operators are internal mechanisms used solely for geometric synthesis.

Given a scaffold that captures the initial design intent, Generator addresses two objectives: *semantic grounding* and *structural exploration*. For **C1**, the generated geometry should preserve the functional semantics encoded by the scaffold, thereby bridging the language–geometry gap. For **C2**, it should explore candidate structures beyond direct retrieval or reproduction of training examples.

To achieve both objectives, we introduce **symbolic-driven latent evolution** in a disentangled latent space. Motivated by programmable metamaterial design, where new structures are constructed by composing and transforming existing motifs (Bastek et al., 2022; Zhao et al., 2025), the proposed method applies symbolic operators to scaffold and initialization representations and evolves the resulting latent variables toward semantically aligned geometries. The disentangled representation further allows structural and semantic factors to be manipulated separately, improving controllability and generation quality. As illustrated in Figure 2, Generator consists of four stages: *initialization*, *disentangled latent encoding*, *symbolic-driven latent evolution*, and *decoding*, which we detail below.

Latent Disentanglement. Existing latent generative frameworks (Xie et al., 2022; Luo et al., 2024a,b) typically use a unified latent space, where different metamaterial attributes, *i.e.*, node positions $\mathbf{P}$, edge connectivity E , lattice vectors $\mathbf{L}$, and properties $\mathbf{y}$, are encoded in a single Gaussian representation. Such entanglement limits controllability and generation quality. To enable finer-grained control and semantic alignment, we propose to disentangle the latent variable. Given a metamaterial $\mathcal{M} = (\mathbf{L}, \mathcal{U})$ with $\mathcal{U} = (\mathbf{P}, E)$ and associated properties $\mathbf{y}$, we decompose the latent variable $\mathbf{z}$ into four components, $\mathbf{z}_l, \mathbf{z}_p, \mathbf{z}_e, \mathbf{z}_s$, corresponding to lattice, node positions, edges, and semantic/property factors, respectively. The inference and generation processes are defined as:

$$\begin{aligned} p_{\theta}(\mathcal{M}, \mathbf{y}, \mathbf{z}) &= p(\mathbf{z}) p_{\theta_l}(\mathbf{L} | \mathbf{z}_l, \mathbf{y}) p_{\theta_p}(\mathbf{P} | \mathbf{z}_p, \mathbf{y}) p_{\theta_e}(E | \mathbf{z}_e, \mathbf{y}) p_{\theta_s}(\mathbf{y} | \mathbf{z}_s), \\ q_{\phi}(\mathbf{z} | \mathcal{M}) &= q_{\phi}(\mathbf{z}_l, \mathbf{z}_p, \mathbf{z}_e, \mathbf{z}_s | \mathcal{M}) \\ &= q_{\phi_1}(\mathbf{z}_l | \mathcal{M}) q_{\phi_2}(\mathbf{z}_p | \mathcal{M}) q_{\phi_3}(\mathbf{z}_e | \mathcal{M}) q_{\phi_4}(\mathbf{z}_s | \mathcal{M}), \\ p(\mathbf{z}) &= p(\mathbf{z}_l) p(\mathbf{z}_p) p(\mathbf{z}_e) p(\mathbf{z}_s), \quad p(\cdot) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{aligned} \quad (1)$$

More details are shown in Appendix D.1. This disentanglement allows Generator to manipulate structural and semantic factors separately, which is critical for scaffold-guided synthesis.

Symbolic-Driven Latent Evolution. To preserve and program the semantics implied by the prompt, Generator should align the latent representation of an initialized structure with that of the scaffold. We focus on two symbolic operators in the main text—*Union* and *Mix*—as they capture two intuitive forms of scaffold-guided programming: structural expansion and semantic interpolation. Additional operators, including *Intersection* (Eq. (13)) and *Negation* (Eq. (15)), are deferred to Appendix B.3.

We first initialize a metamaterial $\mathcal{M}$ from training dataset. *Union* aims to unite the scaffold (represented as $\mathcal{M}'$) with initialized sample. Concretely, we compute a soft node correspondence between $\mathcal{M}$ and $\mathcal{M}'$ using Sinkhorn matching (Frogner et al., 2015) (Eq. (9)), which identifies overlapping components and enables semantic fusion at the node level. *Union* is therefore suitable when the scaffold provides missing structural elements that should be incorporated into the current design.

In contrast, *Mix* performs continuous semantic interpolation between the initialized structure and the scaffold in latent space. Let $p_{\mathcal{M}}(\mathbf{z})$ and $p_{\mathcal{M}'}(\mathbf{z})$ denote their latent distributions. We define

$$p_{\text{mix}}(\mathbf{z} | \lambda_{\text{mix}}) = (1 - \lambda_{\text{mix}}) p_{\mathcal{M}}(\mathbf{z}) + \lambda_{\text{mix}} p_{\mathcal{M}'}(\mathbf{z}), \quad \lambda_{\text{mix}} \in [0, 1], \quad (2)$$

where λ_{mix} controls the degree to which the scaffold influences the resulting design. Smaller values preserve more of the initialization, while larger values pull the latent representation closer to the scaffold semantics. The final expression is shown in Appendix Eq. (11).

With the target latent distributions derived from the symbolic operations, we then propose to evolve the generation latent towards the target latent distribution for two reasons. First, in a disentangled autoencoder, the decoder is trained only on the latent manifold induced by the encoder; operator-based targets may fall outside this manifold and thus lead to invalid decoded structures. Second, symbolic fusion is applied within individual subspaces, disturbing the compatibility among $\mathbf{z}_l, \mathbf{z}_p, \mathbf{z}_e$, and $\mathbf{z}_s$ that the decoder implicitly relies on.

To do so, we evolve the latents of the initialized structure toward the operator-induced target via gradient descent, rather than decoding the target distribution directly. During

optimization, we impose a Sinkhorn-based soft-matching constraint on node and edge distributions to preserve cross-space coherence and keep the latent trajectory close to the learned manifold. Let μ and σ denote Gaussian means and variances. The optimization objective is

$$\begin{aligned}
\mathcal{L}_s &= \text{KL}(\mathcal{N}(\mu_s, \sigma_s) \parallel \mathcal{N}(\mu'_s, \sigma'_s)) \quad (\text{semantic alignment}), \\
\mathcal{L}_{p,e} &= \sum_{k \in \{p,e\}} \sum_{i=1}^{N_M} \sum_{j=1}^{N_{M'}} \Pi_{ij} \text{KL}(\mathcal{N}(\mu_{k,i}, \sigma_{k,i}) \parallel \mathcal{N}(\mu'_{k,j}, \sigma'_{k,j})) \quad (\text{Sinkhorn-weighted alignment}), \\
\mathcal{L}_r &= \sum_{k \in \{p,e\}} \sum_{i \in \{i | r_i < \tau_0\}} \text{KL}(\mathcal{N}(\mu_{k,i}, \sigma_{k,i}) \parallel \mathcal{N}(\mu_{k,i}^{\text{old}}, \sigma_{k,i}^{\text{old}})) \quad (\text{unmatched-node regularization}), \\
\mathcal{L}_{\text{prior}} &= \sum_{i \in \{l,p,e,s\}} \|\mathbf{z}_i\|_2^2 \quad (\text{latent prior regularization}).
\end{aligned} \tag{3}$$

Here, Π_{ij} denotes the Sinkhorn soft-matching matrix between nodes of $\mathcal{M}$ and $\mathcal{M}'$, $\mathcal{N}(\mu_{k,i}^{\text{old}}, \sigma_{k,i}^{\text{old}})$ denotes the pre-optimization distribution, and $\mathcal{N}(\mu', \sigma')$ denotes the operator-induced target. The set $\{i \mid r_i < \tau_0\}$ contains unmatched nodes under Sinkhorn matching (Appendix Alg. 2), with threshold $\tau_0 = 0.1$. More implementation details and mathematical derivations are provided in Appendix B.3.

3.3 Supervisor: Property-Modality Evaluator

Designer and Generator provide language-space reasoning and geometry-space synthesis, respectively, but automated metamaterial discovery still requires efficient feedback in property space. Directly conducting high-fidelity simulation inside the design loop is computationally prohibitive (Lee et al., 2024). Therefore, **Supervisor** is a fast property-space evaluator that provides approximate yet informative feedback for iterative refinement.

Specifically, Supervisor consists of two components as shown in Figure 2: (1) a property predictor trained on structure–property data for rapid mechanical response estimation, and (2) an LLM-based evaluator that jointly considers the design prompt, the candidate structure, and the predicted properties to assess semantic alignment and propose refined feedback. In this way, Supervisor does not replace high-fidelity simulation entirely; instead, it reduces reliance on expensive simulation during the inner optimization loop and improves coordination between language guidance, geometric generation, and property awareness.

Formally, let $\mathcal{V}$ denote the language space, where the human prompt is $V_p \in \mathcal{V}$ and the scaffold generated by Designer is $V_m \in \mathcal{V}$. Let $\mathcal{M}$ denote the geometry space, where Generator maps a latent code $\mathbf{z} \in \mathbb{R}^d$ to a generated structure $\bar{M} \in \mathcal{M}$. We use $f_{\mathcal{V} \rightarrow \mathcal{M}}$ and $f_{\mathcal{M} \rightarrow \mathcal{V}}$ to denote the conversions between scaffold descriptions and geometric representations. Mechanical properties are denoted by $\mathbf{y} \in \mathbb{R}^{d_y}$. Given either a scaffold or a generated structure, Supervisor predicts properties and returns an evaluation signal that drives the two collaboration loops. Algorithm 1 summarizes the Designer–Supervisor collaboration $C_{D/S}$ and the Generator–Supervisor collaboration $C_{G/S}$; more details are provided in Appendix B.2.

Algorithm 1 Designer&Supervisor and Generator&Supervisor Collaborations

Init($\mathbf{z} \mid \mathbf{y}$) represents initializing $\mathbf{z}$ given property $\mathbf{y}$, which is instantiated as a 1-nearest-neighbor (Cover & Hart, 1967) from the dataset followed by Gaussian latent encoding. D, G, S denote Designer, Generator, and Supervisor. S^{pred} and S^{eval} are the predictor and LLM in Supervisor.

$C_{D/S}$ (Scaffold refinement)

Input: prompt V_p ; **Output:** scaffold V_m .

Step 1: $V_m^{(t)} \leftarrow D(V_p^{(t)})$

Step 2: $\mathbf{y}_s^{(t)} \leftarrow S^{\text{pred}}(f_{\mathcal{V}/\mathcal{M}}(V_m^{(t)}))$

Step 3: $\text{score}^{(t)}, V_p^{(t+1)} \leftarrow S^{\text{eval}}(V_m^{(t)}, V_p^{(t)}, \mathbf{y}_s^{(t)})$

Step 4: **repeat** until $\text{score}^{(t)} \geq \tau_{D/S}$

$C_{G/S}$ (Generation refinement)

Input: scaffold V_m ; **Output:** microstructure $\bar{M}$.

Step 1: $\bar{M}^{(t)} \leftarrow G(\mathbf{z}^{(t)}, f_{\mathcal{V}/\mathcal{M}}(V_m)), \mathbf{z}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Step 2: $\mathbf{y}_m^{(t)} \leftarrow S^{\text{pred}}(\bar{M}^{(t)})$

Step 3: $\text{score}^{(t)}, \mathbf{y}_m'^{(t)} \leftarrow S^{\text{eval}}(f_{\mathcal{M}/\mathcal{V}}(\bar{M}^{(t)}), V_p, \mathbf{y}_m^{(t)})$

Step 4: $\mathbf{z}^{(t+1)} \leftarrow \text{Init}(\mathbf{z}^{(t)} \mid \mathbf{y}_m'^{(t)})$ optional reinitialize

Step 5: **repeat** until $\text{score}^{(t)} \geq \tau_{G/S}$

3.4 Multi-Agent Collaboration Mechanism

METASYMBO coordinates Designer, Generator, and Supervisor through two collaboration loops across language, geometry, and property modalities, as illustrated in Figure 3.

In the **Designer-Supervisor loop** ($C_{D/S}$), Designer interprets the input prompt V_p and retrieves a semantically aligned scaffold V_m . Supervisor predicts its mechanical properties and evaluates its consistency with the design intent. If the alignment is insufficient, Supervisor returns a refined prompt V'_p , which guides Designer to retrieve an improved scaffold. This process repeats until satisfactory scaffold-level alignment is reached.

The selected scaffold then initializes the **Generator-Supervisor loop** ($C_{G/S}$). Generator uses the scaffold as a structural prior to synthesize candidate metamaterials $\mathcal{M}$. Supervisor evaluates their predicted properties and prompt alignment, and returns property-aware feedback to guide subsequent generation. Once the evaluation criterion is satisfied, the resulting candidate is passed to higher-fidelity validation, such as finite-element simulation, homogenization, or fabrication.

Importantly, the latent variables and symbolic operators are **internal to Generator**: they transform the retrieved scaffold into new geometries, but do not directly mediate the Designer-Supervisor interaction. Thus, language feedback improves scaffold retrieval, while property feedback guides geometric generation.

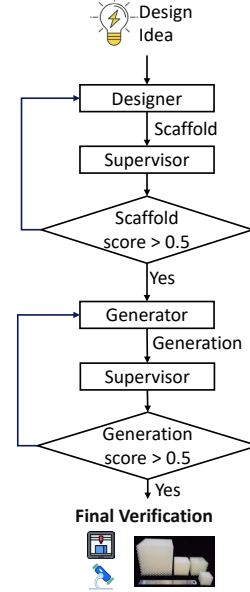


Figure 3: Multi-agent collaboration workflow.

4 Experiments

To evaluate METASYMBO, we investigate four questions: **Q1: Validity.** Does METASYMBO generate metamaterials that are more structurally valid than existing baselines? **Q2: Diversity.** Does METASYMBO generate diverse structures that go beyond the existing design space? **Q3: Language-Guidance Effectiveness.** How effectively do natural-language prompts steer the generation process? **Q4: Operator-based Programmability.** Do the proposed symbolic logic operators enable controllable and meaningful generation? Finally, to further analyze METASYMBO, we ablate the contributions of the latent evolution objectives and study the effectiveness of the disentangled architecture for property prediction.

4.1 Quantitative Comparison

Experimental Setup. We compare METASYMBO against four geometry-aware generative models and six LLM baselines, including two reasoning-oriented LLMs. Experiments are conducted on the MetaModulus dataset (Lumpe & Stankovic, 2021; Chen et al., 2025b) using an 8/2 train/test split, together with 100 metamaterial design prompts from domain experts for prompt-based evaluation. We evaluate performance from three perspectives: (1) *Validity*, measured by symmetry ($\mathcal{V}_S$) and periodicity ($\mathcal{V}_P$); (2) *Diversity*, measured by Coverage Recall (Cov. R.) and Repeat Ratio; and (3) *Language-guidance effectiveness*, measured by a prompt-guidance score produced by an external Supervisor built from GPT-4.1 and a full-data-trained predictor. In the case study, we further validate selected designs via 3D printing as wet-lab testing. More details are provided in Appendix B.4.

Validity. METASYMBO with the Union operator achieves the strongest validity on both symmetry ($\mathcal{V}_S$) and periodicity ($\mathcal{V}_P$). In particular, METASYMBO (GPT-4o-mini, Union) reaches 91.31% on $\mathcal{V}_S$ and 98.35% on $\mathcal{V}_P$, while METASYMBO (Gemini-2.0, Union) follows closely with 89.65% and 95.97%, respectively. By comparison, the strongest generative baseline, CDVAE, attains only 57.03% on $\mathcal{V}_S$ and 0.40% on $\mathcal{V}_P$, while the strongest reasoning LLM baseline, DeepSeek-R, reaches 85.5% and 65.3%. These results answer **Q1** by showing that METASYMBO substantially improves structural validity over both generative and LLM-based baselines.

Approach	$\mathcal{V}_S\% \uparrow$	$\mathcal{V}_P\% \uparrow$	Cov R.% $\uparrow$	Repeat Ratio% $\downarrow$	Prompt Guide score (GPT-4.1) $\uparrow$	Repeat Num. $\downarrow$
<i>Generative Models</i>						
CDVAE (Xie et al., 2022)	57.03	0.40	55.85	N/A	N/A	N/A
DiffCSP (Jiao et al., 2023)	34.46	6.50	95.80	N/A	N/A	N/A
SyMat (Luo et al., 2024b)	41.10	0.00	79.34	N/A	N/A	N/A
Cond-CDVAE Luo et al. (2024a)	19.37	2.00	68.60	N/A	N/A	N/A
<i>LLMs</i>						
GPT-4o-mini (Hurst et al., 2024)	47.29	0.00	73.6	39.66	0.4155	59
Llama-4-maverick (Touvron et al., 2023)	32.06	0.82	65.1	93.39	0.4463	80
Qwen3-235b (Yang et al., 2024)	41.22	4.95	97.1	83.47	0.3820	73
Deepseek-chat (Liu et al., 2024)	46.90	16.53	73.1	85.95	0.4189	65
Gemini-2.0-flash-lite (Team, 2025)	44.31	41.86	27.5	92.56	0.4755	67
Deepseek-Reasoning Guo et al. (2025)	85.5	65.30	86.9	67.7	0.4993	76
METASYMBO (Ablation with Pure LLM Agents)	78.00	29.35	88.9	51.20	0.4832	61
METASYMBO (Gemini2.0, Mix)	64.53	91.74	93.3	0.00	0.5464	0
METASYMBO (GPT-4o-mini, Mix)	76.84	94.17	98.2	0.83	0.5234	0
METASYMBO (Gemini2.0, Union)	89.65	95.97	99.2	10.07	0.4966	56
METASYMBO (GPT-4o-mini, Union)	91.31	98.35	98.7	7.43	0.5531	40

Table 1: Quantitative comparisons. Validity metrics include symmetry ($\mathcal{V}_S\%$) and periodicity ($\mathcal{V}_P\%$). Diversity metrics include coverage recall (Cov R.%), and Repeat Ratio. The prompt guidance metric is denoted as Prompt Guide Score (averaging three runs).

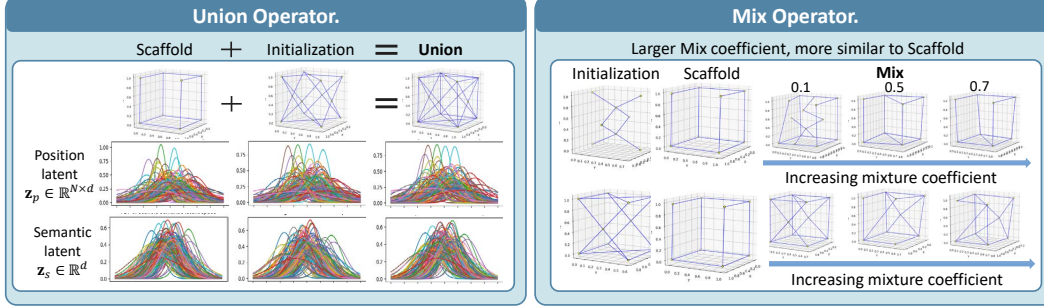


Figure 4: Qualitative analysis shows the proposed operators and latent evolution methods can successfully program the initialized structure towards the semantics of the scaffold.

Diversity. All METASYMBO variants achieve high Coverage Recall ($> 90\%$) on the test set while maintaining low Repeat Ratio ($< 11\%$). This suggests that METASYMBO can discover diverse metamaterials that go beyond the training design space rather than merely reproducing seen patterns. In contrast, LLM baselines exhibit substantially higher repetition and generally lower coverage. Notably, GPT-4o-mini attains a relatively low Repeat Ratio of 39.66%, yet completely fails on periodicity (0% $\mathcal{V}_P$), indicating that low repetition alone does not imply meaningful diversity. Overall, these results provide strong evidence for **Q2**.

Language-Guidance Effectiveness. We measure prompt alignment using the prompt-guidance score from the external Supervisor. METASYMBO (GPT-4o-mini, Union) achieves the highest score of 0.5531, followed by METASYMBO (Gemini-2.0, Mix) at 0.5464. Most METASYMBO variants outperform the LLM baselines, whose scores range from 0.3820 to 0.4993, indicating stronger controllability and better alignment with the intended design semantics. These results answer **Q3** by showing that METASYMBO more effectively translates natural-language intents into geometric designs.

4.2 Model Analysis

Qualitative Analysis. We conduct qualitative analysis to answer **Q4**, focusing on whether symbolic-driven latent evolution enables controllable semantic composition. Figure 4 presents the qualitative analysis of Union and Mix. More details about Intersection and Negation can be found in Appendix C.3. As Figure 4 left shows, Union combines characteristics from both the scaffold and the initialization, which also is reflected in the fused latent distributions in $\mathbf{z}_p$ and $\mathbf{z}_s$. Figure 4 right shows that, as λ_{mix} increases, the generated

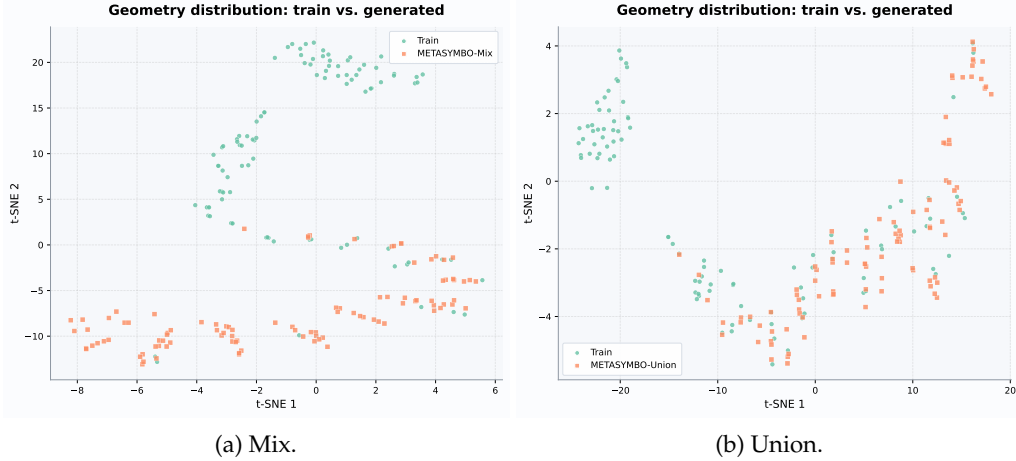


Figure 5: t-SNE visualization of the geometry distributions of training structures and METASYMBO-generated structures using the Mix and Union operators.

structure gradually shifts from the initialization toward the scaffold, demonstrating smooth and controllable interpolation.

Ablation Study. We first ablate geometry and property-space awareness by replacing Generator and Supervisor with pure LLM agents while preserving the collaboration mechanism. As shown in Table 1, this variant remains highly repetitive and yields low validity, highlighting the necessity of geometry-aware generation and property-space supervision. We further ablate each term in Eq. (3) in Table 2. Among them, $\mathcal{L}_{p,e}$ has relatively large influence, showing the effectiveness of Sinkhorn-weighted node/edge alignment in latent evolution. Removing $\mathcal{L}_{prior}$ or $\mathcal{L}_r$ also noticeably hurts validity, as both help preserve feasible structures during evolution. In contrast, $\mathcal{L}_s$ mainly affects the semantic perturbation and thus influences all metrics.

Variant	$\mathcal{V}_S \uparrow$	$\mathcal{V}_P \uparrow$	Cov. R $\uparrow$
w/o $\mathcal{L}_s$	50.9	57.1	93.1
w/o $\mathcal{L}_{p,e}$	47.8	45.7	93.6
w/o $\mathcal{L}_r$	51.6	62.8	94.3
w/o $\mathcal{L}_{prior}$	58.0	62.8	95.1

Table 2: Ablation study.

Generation Distribution Analysis. To analyze whether METASYMBO explores geometries beyond the training distribution, we construct geometric representations using node distances, node count, lattice vectors, and edge lengths, and visualize 1,000 training and generated structures using t-SNE. As shown in Figure 5, generations from both Mix and Union occupy regions that are visibly shifted from the training samples, while also retaining partial overlap with the training distribution. This distributional separation suggests that METASYMBO does not merely reproduce training structures, but explores distinct regions of the geometric design space through symbolic-driven latent evolution.

Property Prediction Performance. We evaluate the introduced property prediction model used in Supervisor. Figure 6 shows the parity plot for Poisson’s ratio, where the high R^2 of 0.877 indicates strong predictive performance. Additional comparisons with other 3D material representation models are provided in Appendix C.4, where the proposed architecture also performs superior.

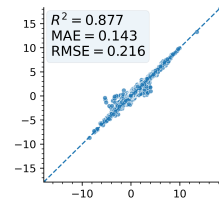


Figure 6: Property prediction.

5 Case Study

Figure 7 illustrates an end-to-end example for designing an auxetic metamaterial with relatively high stiffness. The initial user request specifies only high-level functional intent. In the Designer-Supervisor loop, the prompt is iteratively refined using the retrieved scaffold and its predicted mechanical behavior, resulting in a more explicit structural hypothesis. Importantly, this refinement is geometry- and property-aware rather than a direct language-to-number conversion.

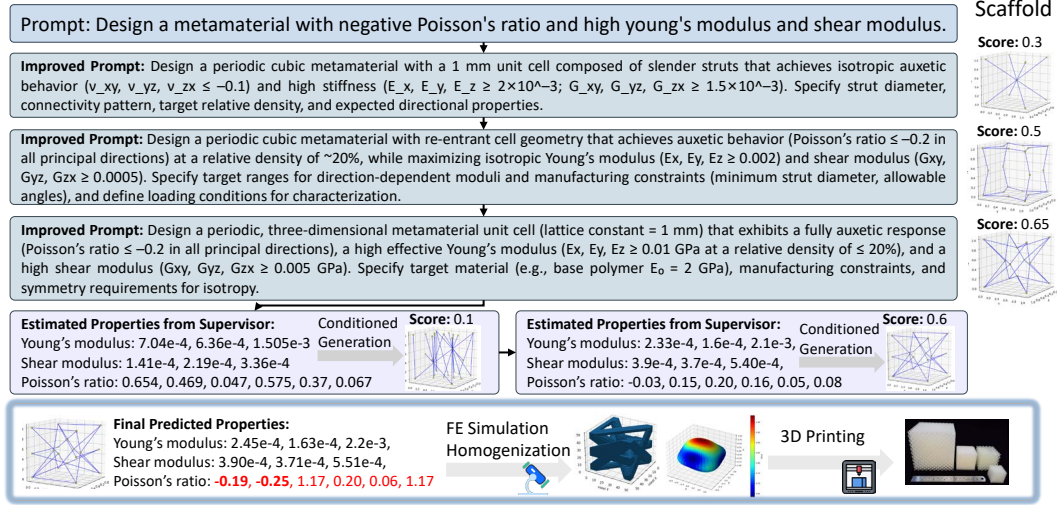


Figure 7: Case study with METASYMBO. FE simulation denotes finite-element simulation.

After a satisfactory scaffold is identified, Generator synthesizes new candidate geometries and Supervisor provides fast property-aware feedback for subsequent generation rounds. The selected design is then evaluated using finite-element simulation and asymptotic homogenization and is subsequently fabricated by 3D printing for physical verification. The resulting structure exhibits predicted negative Poisson's ratios (-0.19 and -0.25) together with relatively high shear-modulus terms, illustrating how METASYMBO can support the transition from qualitative intent to physically testable design hypotheses.

6 Conclusion

In this paper, we present METASYMBO, a multi-agent framework for language-guided metamaterial discovery that bridges language, geometry, and property modalities through coordinated reasoning, generation, and evaluation. Its key technical contribution is **symbolic-driven latent evolution** in a **disentangled latent space**, which enables controllable inference-time refinement beyond simple retrieval or sampling. Across extensive experiments, METASYMBO substantially improves structural validity over strong generative and LLM baselines, while also achieving stronger prompt alignment and maintaining high diversity. The real-world auxetic high-stiffness case study further shows that the framework supports an end-to-end workflow from prompt refinement to practical validation. Overall, our results suggest that language can serve as a practical interface for early-stage metamaterial discovery when coupled with geometric generation and fast property-aware feedback.

Statement of LLM Usage

Large language models (LLMs) were used in this work in three ways: as integral components of our framework (Agent Designer and Agent Supervisor), as baselines for experimental comparison (e.g., GPT-4o-mini, Llama-4-Maverick, Gemini-2.0-flash-lite, Qwen3-235b, Deepseek-chat, Deepseek-Reasoning), and as assistive tools to polish the manuscript's language and presentation; all scientific ideas, methodological contributions, experimental designs, and final claims were conceived and verified by the authors.

Ethics Statement

METASYMBO presents a novel approach to language-guided metamaterial discovery that could significantly enhance the accessibility and efficiency of advanced materials design. By enabling intuitive, language-based, and human-AI collaborative programming over structural generation and property optimization, this framework facilitates metamaterial design for non-experts and accelerates early-stage innovation in domains such as robotics, aerospace, and biomedical engineering. However, as with any generative design tool, there are potential risks. Misuse of the technology, for instance, in the automated design

of materials for harmful or ethically ambiguous applications, could lead to unintended consequences. To mitigate these risks, responsible deployment should include human-in-the-loop validation, interpretability safeguards, and domain-specific oversight. Overall, METASYMBO has the potential to positively transform the scientific design workflow while necessitating thoughtful governance in high-impact use cases.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work is supported by the National Science Foundation under Award No. 2339989, No. 2449769 and No. 2406439, DARPA under contract No. HR00112490370 and No. HR001124S0013, U.S. Department of Homeland Security under Grant Award No. 17STCIN00001-08-00, Amazon-Virginia Tech Initiative for Efficient and Robust Machine Learning, Amazon AWS, Google, Cisco, 4-VA, Commonwealth Cyber Initiative, National Surface Transportation Safety Center for Excellence, and Virginia Tech. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

- 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*, 119(1):e2111505119, 2022.
- Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- Jianpeng Chen, Yujing Wang, Ming Zeng, Zongyi Xiang, Bitan Hou, Yunhai Tong, Ole J Mengshoel, and Yazhou Ren. Customizing graph neural networks using path reweighting. *Information Sciences*, 674:120681, 2024a.
- Jianpeng Chen, Yawen Ling, Jie Xu, Yazhou Ren, Shudong Huang, Xiaorong Pu, Zhifeng Hao, Philip S. Yu, and Lifang He. Variational graph generator for multiview graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11078–11091, 2025a. doi: 10.1109/TNNLS.2024.3524205.
- Jianpeng Chen, Wangzhi Zhan, Haohui Wang, Zian Jia, Jingru Gan, Junkai Zhang, Jingyuan Qi, Tingwei Chen, Lifu Huang, Muhao Chen, et al. Metamatbench: Integrating heterogeneous data, computational tools, and visual interface for metamaterial discovery. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5334–5344, 2025b.
- Jianpeng Chen, Yawen Ling, Yazhou Ren, Shufei Zhang, and Lifang He. Smhgc: Homophily-agnostic multi-view heterophilous graph clustering. *Pattern Recognition*, pp. 113668, 2026.
- Wei Chen, Rachel Sun, Doksoo Lee, Carlos M Portela, and Wei Chen. Generative inverse design of metamaterials with functional responses by interpretable learning. *Advanced Intelligent Systems*, pp. 2400611, 2024b.
- Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.
- 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.
- Nader Engheta and Richard W Ziolkowski. *Metamaterials: physics and engineering explorations*. John Wiley & Sons, 2006.

- Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pp. 2053–2061, 2015.
- Cong Fu, Keqiang Yan, Limei Wang, Wing Yee Au, Michael Curtis McThrow, Tao Komikado, Koji Maruhashi, Kanji Uchino, Xiaoning Qian, and Shuiwang Ji. A latent diffusion model for protein structure generation. In *Learning on Graphs Conference*, pp. 29–1. PMLR, 2024.
- Nate Gruver, Anuroop Sriram, Andrea Madotto, Andrew Gordon Wilson, C. Lawrence Zitnick, and Zachary Ward Ulissi. Fine-tuned language models generate stable inorganic materials as text. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=vN9fpfqoP1>.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.
- Chan Soo Ha, Desheng Yao, Zhenpeng Xu, Chenang Liu, Han Liu, Daniel Elkins, Matthew Kile, Vikram Deshpande, Zhenyu Kong, Mathieu Bauchy, et al. Rapid inverse design of metamaterials based on prescribed mechanical behavior through machine learning. *Nature Communications*, 14(1):5765, 2023.
- Paul Z Hanakata, Ekin D Cubuk, David K Campbell, and Harold S Park. Forward and inverse design of kirigami via supervised autoencoder. *Physical Review Research*, 2(4): 042006, 2020.
- Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model, july 2024. URL <https://www.biorxiv.org/content/10.1101/2024.07.1.v1>, 2024.
- Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 08 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- Yayati Jadhav and Amir Barati Farimani. Large language model agent as a mechanical designer, 2024. URL <https://arxiv.org/abs/2404.17525>, 2024.
- Zian Jia, Fan Liu, Xihang Jiang, and Lifeng Wang. Engineering lattice metamaterials for extreme property, programmability, and multifunctionality. *Journal of Applied Physics*, 127 (15), 2020.
- Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion on lattices and fractional coordinates. In *Workshop on "Machine Learning for Materials" ICLR 2023*, 2023. URL <https://openreview.net/forum?id=VPByphdu24j>.
- Hanxun Jin, Enrui Zhang, Boyu Zhang, Sridhar Krishnaswamy, George Em Karniadakis, and Horacio D Espinosa. Mechanical characterization and inverse design of stochastic architected metamaterials using neural operators. *arXiv preprint arXiv:2311.13812*, 2023.
- Manav Kant, Eric Y Ma, Andrei Staicu, Leonard J Schulman, and Spencer Gordon. Identifiability of product of experts models. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4492–4500. PMLR, 02–04 May 2024.
- Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

- Doksoo Lee, Wei Chen, Liwei Wang, Yu-Chin Chan, and Wei Chen. Data-driven design for metamaterials and multiscale systems: a review. *Advanced Materials*, 36(8):2305254, 2024.
- Xiner Li, Limei Wang, Youzhi Luo, Carl Edwards, Shurui Gui, Yuchao Lin, Heng Ji, and Shuiwang Ji. Geometry informed tokenization of molecules for language model generation, 2025a. URL <https://openreview.net/forum?id=HbZrxBXzks>.
- Zongzhao Li, Jiacheng Cen, Bing Su, Tingyang Xu, Yu Rong, Deli Zhao, and Wenbing Huang. Large language-geometry model: When LLM meets equivariance. In *Forty-second International Conference on Machine Learning*, 2025b. URL <https://openreview.net/forum?id=Qi9rPcdiTY>.
- Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *ICLR*, 2023.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *ICLR*, 2022.
- Thomas S Lumpe and Tino Stankovic. Exploring the property space of periodic cellular structures based on crystal networks. *Proceedings of the National Academy of Sciences*, 118(7):e2003504118, 2021.
- Xiaoshan Luo, Zhenyu Wang, Pengyue Gao, Jian Lv, Yanchao Wang, Changfeng Chen, and Yanming Ma. Deep learning generative model for crystal structure prediction. *npj Computational Materials*, 10(1):254, 2024a.
- Youzhi Luo, Chengkai Liu, and Shuiwang Ji. Towards symmetry-aware generation of periodic materials. *NeurIPS*, 36, 2024b.
- 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.
- Siddharth Narayanan, James D. Braza, Ryan-Rhys Griffiths, Manvitha Ponnampati, Albert Bou, Jon M Laurent, Ori Kabeli, Geemi Wellawatte, Sam Cox, Samuel G Rodrigues, and Andrew White. Aviary: Training language agents on challenging scientific tasks. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025. URL <https://openreview.net/forum?id=25Grz6oh7d>.
- Helda Pahlavani, Kostas Tsifoutis-Kazolis, Mauricio C Saldivar, Prerak Mody, Jie Zhou, Mohammad J Mirzaali, and Amir A Zadpoor. Deep learning for size-agnostic inverse design of random-network 3d printed mechanical metamaterials. *Advanced Materials*, 36(6):2303481, 2024.
- Dilip D Paul. Optical metamaterials: fundamentals and applications, 2010.
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- Henrik Ronellenfitsch, Norbert Stoop, Josephine Yu, Aden Forrow, and Jörn Dunkel. Inverse design of discrete mechanical metamaterials. *Physical Review Materials*, 3(9):095201, 2019.
- Gemini Team. Gemini: A family of highly capable multimodal models, 2025. URL <https://arxiv.org/abs/2312.11805>.

- Jie Tian, Keke Tang, Xianyan Chen, and Xianqiao Wang. Machine learning-based prediction and inverse design of 2d metamaterial structures with tunable deformation-dependent poisson’s ratio. *Nanoscale*, 14(35):12677–12691, 2022.
- Jie Tian, Martin Taylor Sobczak, Dhanush Patil, Jixin Hou, Lin Pang, Arunachalam Ramathan, Libin Yang, Xianyan Chen, Yuval Golan, Xiaoming Zhai, et al. A multi-agent framework integrating large language models and generative ai for accelerated metamaterial design. *arXiv preprint arXiv:2503.19889*, 2025.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Yusong Wang, Tong Wang, Shaoning Li, Xinheng He, Mingyu Li, Zun Wang, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Enhancing geometric representations for molecules with equivariant vector-scalar interactive message passing. *Nature Communications*, 15(1): 313, 2024.
- Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi S. Jaakkola. Crystal diffusion variational autoencoder for periodic material generation. In *ICLR*, 2022. URL https://openreview.net/forum?id=03RLpj-tc_.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- Cheng Zeng, Zulqarnain Khan, and Nathan L Post. Data-efficient and interpretable inverse materials design using a disentangled variational autoencoder. *AI Mater.*, 2025. doi: <https://doi.org/10.55092/aimat20250002>.
- Wangzhi Zhan, Jianpeng Chen, Dongqi Fu, and Dawei Zhou. Unimate: A unified model for mechanical metamaterial generation, property prediction, and condition confirmation. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=X7uVxeFS9A>.
- Jian Zhao, Robert O. Ritchie, and Jian Xiong. High-performance programmable combinatorial lattice materials. *Journal of the Mechanics and Physics of Solids*, 206:106351, 2025. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2025.106351>. URL <https://www.sciencedirect.com/science/article/pii/S0022509625003254>.
- 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, 2023.
- Artem Zholus, Maksim Kuznetsov, Roman Schutski, Rim Shayakhmetov, Daniil Polykovskiy, Sarath Chandar, and Alex Zhavoronkov. Bindgpt: A scalable framework for 3d molecular design via language modeling and reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26083–26091, 2025.

A Detailed Related Works

A.1 Metamaterial Inverse Design

The inverse design of metamaterials involves generating microstructures that achieve user-specified mechanical responses, such as target elastic moduli or Poisson’s ratios. Traditional approaches rely on topology optimization or evolutionary strategies (Deng et al., 2022), but

these methods often struggle with multi-objective formulations and are computationally expensive.

To overcome these limitations, ML models have been introduced to learn structure–property relationships and enable efficient inverse mapping (Ma et al., 2019). Early works apply variational autoencoders (VAEs) (Tian et al., 2022; Pahlavani et al., 2024; Chen et al., 2025a), graph neural networks (GNNs) (Chen et al., 2024a; 2026), or generative adversarial networks (GANs) to model the one-to-many nature of the inverse problem, capturing diverse valid solutions. More recent efforts such as CDVAE (Xie et al., 2022), Cond-CDVAE (Luo et al., 2024a), and SyMat (Luo et al., 2024b) extend this paradigm to periodic and symmetric materials, incorporating physical invariance and conditioning on accurate parameters. DiffCSP (Jiao et al., 2023) furthers this by applying diffusion models over periodic structures with equivariant geometry.

Beyond generative modeling, approaches like Deep-DRAM (Pahlavani et al., 2024) and Cycle-GAN (Tian et al., 2022) offer modular solutions for multi-objective or deformation-dependent inverse design. These frameworks demonstrate size-agnostic predictions, property-aware sampling, and resistance to fatigue or fracture. However, they typically rely on numerical conditioning and do not accept conceptual or language design queries.

Our work addresses this gap by enabling language-conditioned inverse design through multi-agent collaborations. Compared to prior methods that require exact target vectors, our framework supports symbolic and language prompts and incorporates multi-modality agents that operate over geometry, physics properties, and language. This design philosophy enables more interpretable, flexible, and interactive design workflows.

A.2 Agentic AI for Material Discovery

Recent advances in LLMs have shown remarkable potential in augmenting scientific and engineering workflows through reasoning, planning, and symbolic manipulation. In the metamaterial domain, where design spaces are combinatorially vast and high-fidelity evaluation is computationally expensive, LLMs offer a promising interface for intuitive, high-level control of generative pipelines.

Several works integrate LLMs with physical constraints for 3D material generation. For example, ESM3 Hayes et al. (2024) jointly models sequence, structure, and function via tokenized multimodal prompts; BindGPT Zholus et al. (2025) generates 3D molecules via a language model trained on spatial data; CrystalLLM (Gruver et al., 2024) fine-tunes LLMs to generate inorganic crystal structures as text; EquiLLM (Li et al., 2025b) combines LLMs with equivariant GNNs to model physical systems; Geo2Seq (Li et al., 2025a) introduces SE(3)-invariant tokenization for LLMs to generate 3D molecules. However, these works based on tokenization of geometric structures, have limited exploration in geometric space. MetaSymbO similarly spans modalities, but uniquely enables inference-time symbolic composition and extrapolative design beyond training distributions.

In addition, the latest related works, such as CrossMatAgent (Tian et al., 2025), demonstrate multi-agent systems that couple LLMs with visual generation models and physics-based simulators to automate design tasks. These systems enable LLMs to act as supervisory agents that propose structures, query simulations, and refine designs, leveraging capabilities from models like GPT-4o (Hurst et al., 2024), DeepSeek (Liu et al., 2024), Gemini (Team, 2025), and Qwen2.5 (Yang et al., 2024) for multimodal understanding and structured reasoning.

Complementary research (Jadhav & Farimani, 2024) frames LLMs as autonomous mechanical designers capable of iteratively generating and refining truss structures via feedback from finite element analysis (FEA), achieving performance competitive with traditional optimization methods. These findings support the feasibility of deploying LLMs in complex inverse design settings without task-specific training.

However, challenges remain: LLMs often lack geometric awareness and are not pretrained on physical design tasks. To bridge this gap, our approach introduces LLMs as language agents operating in a cooperative multi-agent setting. Rather than relying on zero-shot

generation alone, our system aligns language-derived intents with latent structural priors and property constraints, amplifying LLM utility via multimodal feedback loops.

By combining the natural abstraction power of LLMs with geometry-aware agents and predictive supervision, we demonstrate that LLMs can serve not only as query interfaces but as active participants in the multi-agent collaboration framework.

B Implementation Details

B.1 Details of Prompt for Agents Instantiation

To instantiate the goals as we discussed in Section 3, we provide the detailed prompts for Agent Designer and Agent Supervisor to deal with the inputs, outputs, and constraints. Figures 8 and 9 illustrate the prompt for the designer and supervisor. Specifically, Designer highlights *locate* from existing literature for a simple structure containing semantics similar to the input text. Alternatively, the supervisor emphasizes utilizing not only the literature but also the predicted mechanical properties to obtain the score, as well as an improved prompt that will provide feedback to designer for the next design iteration.

Agent Designer Prompt
You are a metamaterial scientist specializing in structural design and mechanical characterization. You have expert knowledge of canonical 3-D architectures (octet-truss, BCC, SC, Kelvin cell, Diamond, TPMS, etc.) and their typical mechanical responses. Task ----- Given a single <i>*design requirement*</i>, locate in the metamaterial literature the simplest existing basic substructure (motif) that meets the requirement. Describe this motif as an undirected graph: - **Nodes** — 3-D fractional coordinates. - **Edges** — pairs of node indices. Output the graph in a code block exactly as shown below; provide **no additional text, commentary, or reasoning**. Input ----- Design prompt (free text). Output format ----- ~~~ Node number: <N> Node coordinates: (x0, y0, z0) ... (xN-1, yN-1, zN-1) Edges: (i0, j0) ... (iM-1, jM-1) ~~~ Constraints ----- - Keep the motif as simple as possible (minimal nodes/edges). - Return the output <i>*only*</i> in the specified layout and code-block format. - Do not include any other information.

Figure 8: Prompt for Designer.

Agent Supervisor Prompt
You are a metamaterial scientist specializing in structural design and mechanical characterization. You are fluent in the geometry and typical property ranges of canonical 3-D architectures such as octet-truss, BCC, SC, Kelvin cell, Diamond, and TPMS families. Task ----- Given one <i>*design prompt*</i> and a corresponding <i>*metamaterial structure*</i> with its mechanical properties (Young's modulus, Shear modulus, and Poisson's ratio), output: **Score** — a single real number in **[0, 1]** evaluating how well the structure and the provided properties (if have) fulfills the design prompt (0 = poor, 1 = perfect). **Improved Prompt** — the original design requirement rewritten with clearer, more specific engineering details. **Predicted Properties** — your best estimate of the structure's mechanical response: *Young's modulus* (Ex, Ey, Ez) *Shear modulus* (Gxy, Gyz, Gzx) *Poisson ratio* (vxy, vyx, vxz, vzx, vyz, vzy) Input Format ----- Prompt: <free-text design requirement> Structure: ~~~ Node number: <N> Node coordinates: (x1, y1, z1) ... (xN, yN, zN) Edges: (i0, j0) ... (iM, jM) ~~~ Lattice lengths: [a, b, c] Lattice angles: [α, β, γ] Properties: Young's modulus: [Ex, Ey, Ez] Shear modulus: [Gxy, Gyz, Gzx] Poisson ratio: [vxy, vyx, vxz, vzx, vyz, vzy] Output Format ----- Score: <float in [0,1]> Improved Prompt: <refined design requirement> Improved Properties: Young's modulus: [Ex, Ey, Ez] Shear modulus: [Gxy, Gyz, Gzx] Poisson ratio: [vxy, vyx, vxz, vzx, vyz, vzy] Constraints ----- - Return <i>*only*</i> the fields specified above, in the exact order and layout. - Provide no additional commentary, explanations, or reasoning steps. - For mechanical properties, their value has scales: Ex, Ey, Ez in [0, 1e-2]; Gxy, Gyz, Gzx in [0, 1e-2]; vxy, vyx, vxz, vzx, vyz, vzy in [-20, +20]. ~~~~~

Figure 9: Prompt for Supervisor.

B.2 Details of Collaboration Mechanisms with Supervisor

Algorithm 1 shows the collaboration steps of Agent 1&3 collaboration and Agent 2&3 collaboration. To clarify, we further introduce the collaboration steps as follows in detail.

Agent Designer&Supervisor Collaboration. Agent Designer receives the prompt V_p from human and outputs the best-matched basic structures (*i.e.*, scaffold) $V_m = A_1(V_p)$ by exploiting its literature base. After that, the scaffold is fed into Agent Supervisor, which first predicts the associated mechanical properties $\mathbf{y}_s = A_3^{\text{pred}}(V_m)$. Based on both the predicted properties $\mathbf{y}_s$ and the original input-output pairs (V_p, V_m) , Agent Supervisor computes a match score and provides an improved prompt $s, V'_p = A_3^{\text{eval}}(V_m, V_p, \mathbf{y}_s)$. Finally, it returns score s and improved prompt V'_p to Agent Designer. This loop repeats till a good evaluation score is received. Formally, using t to denote iteration and $\tau_{D/S}$ be threshold.

Agent Generator&Supervisor Collaboration. Agent Generator begins by initializing a latent Gaussian noise vector or randomly selects an initialization from dataset, which serves as the starting point in the generation process, denoted as $\bar{M} = A_2(\mathbf{z})$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The generated structure $\bar{M}$ is then passed to Agent Supervisor to predict its mechanical properties as $\mathbf{y}_m = A_3^{\text{pred}}(\bar{M})$. Subsequently, Agent Supervisor evaluates the quality of the structure by computing a match score s and generating an updated property target $\mathbf{y}'_m$ using the evaluation function A_3^{eval} . The updated properties $\mathbf{y}'_m$ are then (optionally) used to guide the initialization of the latent vector $\mathbf{z}$ for the next iteration in Agent Generator. This iterative process continues until the evaluation score s reaches the threshold $\tau_{G/S}$ as shown in the right column of Alg. 1.

B.3 Details of Symbolic-Driven Latent Evolution

In this section, we detail the process in Generator, especially the disentangled generation and symbolic operators.

Disentangling Latent Generation. Figure 10 describes the implementation of the encoder and decoder for latent space construction. Considering the complete metamaterial representations $\mathcal{M} = (\mathbf{L}, \mathcal{U})$ and $\mathcal{U} = (\mathbf{P}, E)$, with four representation dimension, *i.e.*, lattice vector $\mathbf{L}$, associated property $\mathbf{y}$, node positions $\mathbf{P}$ and edges E , we disentangle the latent $\mathbf{z}$ to $\mathbf{z}_l, \mathbf{z}_p, \mathbf{z}_e, \mathbf{z}_s$ implying the lattice, coordinate position, edges, and metamaterial semantics (referring to the mechanical properties) with Gaussian prior:

$$\begin{aligned} q_\phi(\mathbf{z} | M) &= q_\phi(\mathbf{z}_l, \mathbf{z}_p, \mathbf{z}_e, \mathbf{z}_s | M) = q_{\phi_1}(\mathbf{z}_l | M) q_{\phi_2}(\mathbf{z}_p | M) q_{\phi_3}(\mathbf{z}_e | M) q_{\phi_4}(\mathbf{z}_s | M), \\ p(\mathbf{z}) &= p(\mathbf{z}_l)p(\mathbf{z}_p)p(\mathbf{z}_e)p(\mathbf{z}_s), \text{ where each } p(\cdot) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{aligned} \quad (4)$$

Therefore, the metamaterial-specific decoder reconstruct the full metamaterial spaces $\mathcal{M}$ using conditional likelihood:

$$p_\theta(M | \mathbf{z}) = p_{\theta_1}(\mathbf{L} | \mathbf{z}_l) p_{\theta_2}(\mathbf{P} | \mathbf{z}_p) p_{\theta_3}(E | \mathbf{z}_e) p_{\theta_4}(\mathbf{y} | \mathbf{z}_s), \quad (5)$$

which imposes the four latent spaces with specific physical meanings. Finally, the derived ELBO for the disentangled VAE can be expressed as:

$$\begin{aligned} \mathcal{L}_{\text{ELBO}}(\theta, \phi; \mathcal{M}) &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathcal{M})} [\log p_{\theta_1}(\mathbf{L} | \mathbf{z}_l) + \log p_{\theta_2}(\mathbf{P} | \mathbf{z}_p) + \log p_{\theta_3}(E | \mathbf{z}_e) + \log p_{\theta_4}(\mathbf{y} | \mathbf{z}_s)] \\ &\quad - \text{KL}(q_{\phi_1}(\mathbf{z}_l | \mathcal{M}) \| p(\mathbf{z}_l)) - \text{KL}(q_{\phi_2}(\mathbf{z}_p | \mathcal{M}) \| p(\mathbf{z}_p)) \\ &\quad - \text{KL}(q_{\phi_3}(\mathbf{z}_e | \mathcal{M}) \| p(\mathbf{z}_e)) - \text{KL}(q_{\phi_4}(\mathbf{z}_s | \mathcal{M}) \| p(\mathbf{z}_s)) \end{aligned} \quad (6)$$

Similarly, we can derive the objective for disentangled LDM with the AE (Eqs. 4 and 5) as follows:

$$\mathcal{L}_{\text{LDM}}^{(t)}(\theta) = \sum_{i \in \{l, p, e, s\}} \mathbb{E}_{\mathbf{z}_i^{(0)}, \epsilon_i, t} \left[\left\| \epsilon_i - \epsilon_{\theta_i}(\mathbf{z}_i^{(t)}, t) \right\|^2 \right]. \quad (7)$$

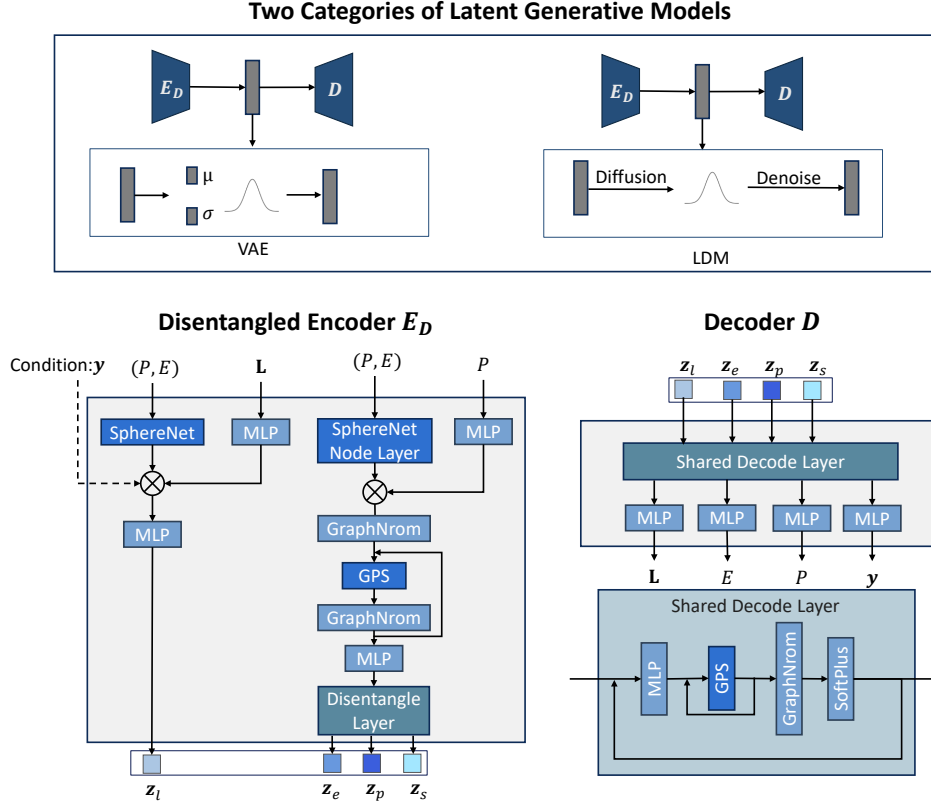


Figure 10: Implementation details of the disentangled encoder and decoder. After the geometry is encoded to a disentangled latent Gaussian space, it is trivial to implement VAEs and DMs for generation as the upper part shown. Please find more details in our codebase.

Note that the four encoder parameters $\{\phi_i\}_{i=1}^4$ and decoder/denoising parameters $\{\theta_i\}_{i=1}^4$ contain shared layers, and only the final head layers need to be fine-tuned for disentangling. In addition, we can implement conditional generation by concatenating condition y_{cond} in the decoder/denoising model as $p_\theta(M | \mathbf{z}, y_{cond})$ or $\epsilon_\theta(\mathbf{z}, t, y_{cond})$.

With this design, we are able to control the generation process regarding all four aspects of a metamaterial in four latent spaces, respectively, achieving a fine-grained generation control.

Symbolic Logic Operators. We introduce four symbolic logic operators for the scaffold guidance, *i.e.*, Union, Mix, Intersection, and Negation.

Union aims to expand the node set of the source metamaterial M according to the guidance scaffold M' in node level. More than a simple expansion of node set, it further fuses the semantics in semantic space. Formally, let $\mathbf{z}_i \sim \mathcal{N}(\mu_i, \Sigma_i)$, $i \in \mathcal{I}_M = \{1, \dots, N_M\}$ and $\mathbf{z}'_j \sim \mathcal{N}(\mu'_j, \Sigma'_j)$, $j \in \mathcal{I}'_M = \{1, \dots, N_{M'}\}$ be the node-level latents of original M and scaffold M' , where N_M and $N_{M'}$ denotes their node numbers. We introduce a differentiable soft matching algorithm–Sinkhorn transport (Frogner et al., 2015) that measures the probability mass of two distributions being regarded as the same physical node—for differentiable assignment matrix (where differentiability enables gradient-based latent optimization):

$$\mathbf{P} = [\mathbf{P}_{ij}]_{i \in \mathcal{I}_M, j \in \mathcal{I}'_M} \in [0, 1]^{N_M \times N_{M'}}. \quad (8)$$

Therefore, $r_i = \sum_j \mathbf{P}_{ij}$, $c_j = \sum_i \mathbf{P}_{ij}$, $0 \leq r_i, c_j \leq 1$ gives the probability of overlapped nodes in M and M' , respectively, and $\rho = \sum_i r_i = \sum_j c_j$ representing the number of overlapping nodes. The detailed computation of the Sinkhorn matrix $\mathbf{P}$ is illustrated in Algorithm 2.

Algorithm 2 SINKHORN_LOG: Log-Stabilised Sinkhorn Iteration

Require: Cost matrix $C \in \mathbb{R}^{N_s \times N_t}$, entropic weight $\varepsilon > 0$, maximum iterations T , tolerance τ

Ensure: Transport plan $P \in \mathbb{R}^{N_s \times N_t}$

```

1:  $K \leftarrow -\frac{C}{\varepsilon}$  ▷ kernel in the log domain
2:  $f \leftarrow \mathbf{0}_{N_s}; g \leftarrow \mathbf{0}_{N_t}$ 
3: for  $t = 1, 2, \dots, T$  do
4:    $f_{\text{prev}} \leftarrow f$ 
5:   /* row scaling update */
6:    $\tilde{K} \leftarrow K + g^\top$  ▷  $\tilde{K}_{ij} = K_{ij} + g_j$ 
7:    $\tilde{K} \leftarrow \tilde{K} - \max_j \tilde{K}_{ij}$  ▷ row-wise stabilisation
8:    $f \leftarrow -\text{LSE}_j(\tilde{K}_{ij})$ 
9:   /* column scaling update */
10:   $\hat{K} \leftarrow K + f$  ▷  $\hat{K}_{ij} = K_{ij} + f_i$ 
11:   $\hat{K} \leftarrow \hat{K} - \max_i \hat{K}_{ij}$  ▷ column-wise stabilisation
12:   $g \leftarrow -\text{LSE}_i(\hat{K}_{ij})$ 
13:   $\delta \leftarrow \|f - f_{\text{prev}}\|_\infty$ 
14:  if  $\delta > 10^4$  then ▷ numerical blow-up guard
15:    break
16:  end if
17:  if  $\delta < \tau$  then ▷ convergence reached
18:    break
19:  end if
20: end for
21:  $P \leftarrow \exp(K + f + g^\top)$ 
22: return  $\hat{P}$ 

```

Using $\delta_{(\mu, \Sigma)}$ denotes Dirac measure centered at the parameter pair (μ, Σ) , which indicates one node in the union has distribution $\mathcal{N}(\mu, \Sigma)$, we can construct the discrete measure of Union on continuous Gaussian space as $\pi_\cup$ in Eq. (9) for gradient-based optimization.

$$\pi_\cup = \frac{\mu_\cup}{Z}, \text{ where } Z = N_M + N_{M'} - \rho, \text{ and}$$

$$\mu_\cup = \underbrace{\sum_{i \in \mathcal{I}_M} (1 - r_i) \delta_{(\mu_i, \Sigma_i)}}_{\text{nodes unique to } M} + \underbrace{\sum_{j \in \mathcal{I}_{M'}} (1 - c_j) \delta_{(\mu'_j, \Sigma'_j)}}_{\text{nodes unique to } M'} + \underbrace{\sum_{i,j} P_{ij} \delta_{(\mu_i, \Sigma_i)}}_{\text{overlapping nodes, preserving } M} \quad (9)$$

Mix. Unlike Union, the Mix operator blends the latent distributions of the original metamaterial M and the scaffold M' into a single composite distribution, where the contribution of the scaffold is modulated by a guidance coefficient $\lambda_{\text{mix}} \in [0, 1]$. Its probabilistic form is expressed as:

$$p_{\text{mix}}(\mathbf{z} \mid \lambda_{\text{mix}}) = (1 - \lambda_{\text{mix}})p_M(\mathbf{z}) + \lambda_{\text{mix}}p_{M'}(\mathbf{z}), \quad (10)$$

where p_M and $p_{M'}$ denote the empirical latent distributions induced by M and M' respectively. However, Eq. (10) is typically intractable due to the complexity of p_M and $p_{M'}$. Considering that latent distributions are Gaussian, we adopt a simplified moment-matching approximation (Bishop & Nasrabadi, 2006):

$$p_{\text{mix}}(\mathbf{z} \mid \lambda_{\text{mix}}) \approx \mathcal{N}\left(\mathbf{z}; (1 - \lambda_{\text{mix}})\boldsymbol{\mu}_M + \lambda_{\text{mix}}\boldsymbol{\mu}_{M'}, \text{diag}((1 - \lambda_{\text{mix}})\boldsymbol{\sigma}_M + \lambda_{\text{mix}}\boldsymbol{\sigma}_{M'})^2\right) \quad (11)$$

Intersection operator aims at identifying the common semantics or overlapping components between the two distributions of M and M' in the latent space. To do so, we introduce Product-of-Expert (PoE), which results in the distribution focusing on regions of high

probability shared by both p_M and $p_{M'}$. This can effectively sharpen the distribution, making generated samples focus more on shared structure features. Formally, the Intersection of two distribution using PoE is:

$$p_{\text{int}}(\mathbf{z}) \propto p_M(\mathbf{z}) \cdot p_{M'}(\mathbf{z}). \quad (12)$$

Considering both $p_M(\mathbf{z})$ and $p_{M'}(\mathbf{z})$ hold Gaussian prior, the resulting Intersection distribution can be derived as:

$$\begin{aligned} p_{\text{int}}(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\text{int}}, \boldsymbol{\Sigma}_{\text{int}}), \\ \text{where } \boldsymbol{\Sigma}_{\text{int}} &= (\boldsymbol{\Sigma}_M^{-1} + \boldsymbol{\Sigma}_{M'}^{-1})^{-1}, \text{ and } \boldsymbol{\mu}_{\text{int}} = \boldsymbol{\Sigma}_{\text{int}}(\boldsymbol{\Sigma}_M^{-1}\boldsymbol{\mu}_M + \boldsymbol{\Sigma}_{M'}^{-1}\boldsymbol{\mu}_{M'}) \end{aligned} \quad (13)$$

Negation is contrary to Intersection that emphasizes common high-density regions, Negation aims to suppress the influence of high-density regions in the latent space of M' from that of M . Accordingly, we can define the unnormalized probability density as:

$$p_{\text{neg}}(\mathbf{z}) \propto \frac{p_M(\mathbf{z})^\alpha}{p_{M'}(\mathbf{z})^\beta}, \quad (14)$$

where $\alpha, \beta > 0$ are hyperparameters that respectively control the strength of preservation and suppression.

However, under this construction, the resulting distribution $p_{\text{neg}}(\mathbf{z})$ is no longer strictly Gaussian, potentially leading collapse of decoding. Therefore, we perform moment-matching to approximate it as a single Gaussian, as shown in Eq. (15).

$$\begin{aligned} p_{\text{neg}}(\mathbf{z}) &\approx \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\text{neg}}, \boldsymbol{\Sigma}_{\text{neg}}), \\ \text{where } \boldsymbol{\Sigma}_{\text{neg}}^{-1} &= \alpha\boldsymbol{\Sigma}_M^{-1} - \beta\boldsymbol{\Sigma}_{M'}^{-1}, \text{ and } \boldsymbol{\mu}_{\text{neg}} = \boldsymbol{\Sigma}_{\text{neg}} \left(\alpha\boldsymbol{\Sigma}_M^{-1}\boldsymbol{\mu}_M - \beta\boldsymbol{\Sigma}_{M'}^{-1}\boldsymbol{\mu}_{M'} \right) \end{aligned} \quad (15)$$

Here, α and β are hyperparameters that control the strength of preservation and negation, respectively. A larger β increases the degree of suppression exerted by the latent distribution of M' .

Gaussian Latent Evolution. Although a symbolic-logic operator yields a closed-form target Gaussian, decoding from that distribution directly poses two problems. (1) In a disentangled AE the decoder is trained only on the latent manifold induced by the encoder. Closed-form operations such as *Mixture*, *Intersection*, or *Negation* can push the target distribution far outside this manifold, so the decoder may produce implausible geometries. (2) Symbolic operators act component-wise and therefore fuse two latents within the same sub-space; statistical dependencies across the four disentangled sub-spaces vanish, breaking the compatibility that the decoder relies on.

To resolve both issues, we *optimize the original latent vector* toward the closed-form target by gradient descent. During this process, we impose a Sinkhorn-based soft-matching loss on node and edge distributions, which preserves cross-space coherence and keeps the trajectory on the learned manifold. Formally, the latent optimization loss can be the weighted sum of Eq. (16). In detail, KL between the semantic distribution $\mathcal{L}_s$ is to learn targeted semantics; Sinkhorn weighted KL between the node position/edge distributions $\mathcal{L}_{p,e}$ is to learn node alignment in both edge and node space; regularization $\mathcal{L}_r$ is to preserve original distributions of non-overlapping nodes; and latent prior ℓ_2 norm $\mathcal{L}_{\text{prior}}$ is to prevent distribution drift.

$$\begin{aligned} \mathcal{L}_s &= \text{KL}(\mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s) || \mathcal{N}(\boldsymbol{\mu}'_s, \boldsymbol{\sigma}'_s)) \quad (\text{Graph-level semantic optimization}), \\ \mathcal{L}_{p,e} &= \sum_{k \in \{p,e\}} \sum_{i=1}^{N_M} \sum_{j=1}^{N_{M'}} \mathbf{P}_{ij} \text{KL}(\mathcal{N}(\boldsymbol{\mu}_{k,i}, \boldsymbol{\sigma}_{k,i}) || \mathcal{N}(\boldsymbol{\mu}'_{k,j}, \boldsymbol{\sigma}'_{k,j})) \quad (\text{Node-level pos./edge alignment}), \\ \mathcal{L}_r &= \sum_{k \in \{p,e\}} \sum_{i \in \{i | r_i < \tau_o\}} \text{KL}(\mathcal{N}(\boldsymbol{\mu}_{k,i}, \boldsymbol{\sigma}_{k,i}) || \mathcal{N}(\boldsymbol{\mu}_{k,i}^{\text{old}}, \boldsymbol{\sigma}_{k,i}^{\text{old}})) \quad (\text{Node-level regularization}), \\ \mathcal{L}_{\text{prior}} &= \sum_{i \in \{l,e,p,s\}} \|\mathbf{z}_i\|_2^2 \quad (\text{Prior regularization}). \end{aligned} \quad (16)$$

Here, $\mathcal{N}(\mu_{k,i}^{old}, \sigma_{k,i}^{old})$ denotes the original latent distribution before optimization, $\mathcal{N}(\mu', \sigma')$ represents computed target distribution, and $\{i | r_i < \tau_o\}$ denotes the alone nodes in M as computed in Eq. (8), where $\tau_o = 0.1$.

B.4 Details of Experimental Setups

B.4.1 Baselines

We compare four material generative models and six LLMs, including two reasoning-focused LLMs.

Generative Models.

- CDVAE (Xie et al., 2022): A variational autoencoder (VAE)-based model for crystal generation, imposing periodic boundary constraints to capture lattice invariance. It encodes both fractional coordinates and lattice vectors, enabling physically valid crystalline outputs.
- DiffCSP (Jiao et al., 2023): A diffusion model (DM) that incorporates $SE(3)$ -equivariant constraints over lattices and fractional coordinates, ensuring rotational and translational invariance during crystal structure prediction.
- SyMat (Luo et al., 2024b): A VAE-based framework with symmetry-aware constraints that explicitly enforce geometric symmetry in periodic metamaterials, thereby improving validity under symmetry-preserving transformations.
- Cond-CDVAE (Luo et al., 2024a): A conditional extension of CDVAE that integrates property vectors into the generative process, enabling structure generation conditioned on target physical properties while preserving periodicity.

Large Language Models (LLMs).

- GPT-4o-mini (Hurst et al., 2024): A lightweight multimodal variant of GPT-4, optimized for efficiency (tens of billions of parameters), capable of text-vision understanding but not a deep-thinking model.
- Llama-4-maverick (Touvron et al., 2023): An open-weight LLM with around 70B parameters, trained for general reasoning and generation. It is not specialized as a reasoning model but provides balanced accuracy and efficiency.
- Deepseek-chat (Liu et al., 2024): A chat-optimized conversational LLM (hundreds of billions of parameters) designed for dialogue and general problem solving; not a dedicated reasoning model.
- Qwen3-235b (Yang et al., 2024): A 235B-parameter deep-thinking model with chain-of-thought style reasoning abilities, explicitly optimized for multi-step reasoning and complex scientific problem solving.
- Deepseek-Reasoning Guo et al. (2025): A reasoning-specialized variant of Deepseek, trained with reinforcement learning to enhance long-chain reasoning. It belongs to the emerging class of deep-thinking LLMs.
- Gemini-2.0-flash-lite (Team, 2025): A highly efficient multimodal LLM from Google’s Gemini family (tens of billions of parameters), designed for fast inference across text, vision, and structured data, not explicitly reasoning-focused.

B.4.2 Metrics

To evaluate the performance, we first employ validity from two aspects, *i.e.*, symmetries and periodicity (Luo et al., 2024b;a). To evaluate the generation diversity, we conduct coverage recall that measures how many structures in the test dataset are covered by the generated structures (Chen et al., 2025b). In addition, we introduce the repeat ratio to indicate how many of the same structures are generated by one model. The more detailed computation of these metrics is illustrated as follows.

Symmetry Validity. Symmetry validity is to evaluate the symmetry of a structure by computing the central symmetry ratio ($\mathcal{V}_S$) of a graph in the 3D Cartesian space. Specifically, $\mathcal{V}_S$ is defined as:

$$\mathcal{V}_S = \frac{1}{N_L} \sum_k \frac{N_{S_k} \cdot \sum_i^{N_k} s_{degree_i}}{N_k^2}, \quad (17)$$

where N_L is the number of generated structures, N_k is the node number of k -th structure, and N_{S_k} is the number of Symmetrical Node that is defined in Definition 1 in k -th structure, and s_{degree_i} denotes Symmetry Degree that is defined in Definition 2. In detail, we define a symmetrical node as a node that can find central symmetrical ones within an error range:

Definition 1 (Symmetrical Node). $\mathbf{p}_c$ denotes central coordinates in this structure, and ϵ is a positive hyperparameter. We consider node i with coordinates $\mathbf{p}_i$ to be a symmetrical node iff exists another node j in the structure satisfies: $\|\mathbf{p}_i + \mathbf{p}_j - 2\mathbf{p}_c\|_2 < \epsilon$.

In addition, the symmetry degree of a node is defined as the error value of the corresponding "most symmetric" node pair divided by the distance between the central coordinates and the farthest node.

Definition 2 (Symmetry Degree). $\mathbf{p}_c$ denotes central coordinates in this structure, and j is a node in this structure. The symmetry degree of node i in a structure is defined as: $s_{degree_i} = \frac{\epsilon_{max} - s_{error_i}}{\epsilon_{max}}$, where $\epsilon_{max} = \max_j \|\mathbf{p}_c - \mathbf{p}_j\|_2$, and $s_{error_i} = \min_j \|\mathbf{p}_i + \mathbf{p}_j - 2\mathbf{p}_c\|_2$.

Periodicity Validity. According to Definition 3, a lattice is formed by periodically repeating unit cell structures along the lattice vectors $\mathbf{L}$. Therefore, the periodicity, denoted as $\mathcal{V}_P$, aims to assess the generated structures at the lattice level. This metric aims to evaluate whether the structures can repeat for constructing a lattice. Formally, we define the necessary condition of periodicity of a structure:

Definition 3 (Periodicity). Given a structure with node positions $\mathbf{P}$ and lattice vectors $\mathbf{L}$, for each dimension $d \in \{0, 1, 2\}$, there exist at least one pair of coordinate points $\mathbf{p}_i$ and $\mathbf{p}_j$ such that $\mathbf{p}_i + \mathbf{l}_d$ is approximately equal to $\mathbf{p}_j$ in the L1 norm within a tolerance range ϵ . Formally,

$$\begin{aligned} \forall d \in \{0, 1, 2\}, \\ \exists i \in \{0, 1, \dots, N-1\}, \exists j \in \{0, 1, \dots, N-1\}, \\ \text{s.t. } \|(\mathbf{c}_i + \mathbf{l}_d) - \mathbf{c}_j\|_1 < \epsilon. \end{aligned}$$

Eventually, the evaluation of the periodicity of generated lattices can be computed by $\mathcal{V}_P = \frac{N_P}{N_L}$, where N_P denotes the number of generated structures that satisfy Definition 3.

Coverage Recall (Cov. R). Intuitively, coverage recall measures how many structures in the ground truth dataset are covered by generated structures, i.e.,

$$\text{COV}_R = \frac{1}{N_t} |\{i \in [1, \dots, N_t] : \exists k \in [1, \dots, N_L], D(\mathbf{P}_i^*, \mathbf{P}_k) < \epsilon_{cov}\}|, \quad (18)$$

where $D(\mathbf{P}_i^*, \mathbf{P}_k)$ is a distance metrics to evaluate the distance between i th structure and j th structure.

Repeat Ratio. In detail, for all generated structures, we compute their distance $D(\mathbf{P}_i^*, \mathbf{P}_k)$ and regard the $D(\mathbf{P}_i^*, \mathbf{P}_k) < \epsilon$ as matched structures. To avoid repeat count, we constrain that each structure matches once.

In addition, for the validity and diversity of LLMs that require a prompt input, we use the same prompt of LLMs and ours as "Design a valid and diverse structure, ensure it satisfies symmetry, and periodicity".

Design Prompts
Design a structure with high stiffness.
Create a metamaterial with negative Poisson’s ratio (auxetic).
Generate a structure with ultra-lightweight and moderate stiffness.
Optimize a design for maximum load-bearing capacity.
Design a structure that is extremely flexible in one direction but rigid in the orthogonal direction.
Build a material with a specific Young’s modulus value (e.g., 100 MPa).
Maximize shear strength while minimizing density.
Create a structure with directionally dependent compressive strength.
Design a metamaterial that behaves like a spring under compression.
Create a structure that can absorb large impacts without permanent deformation.

Table 3: 10 samples from all 100 design prompts.

B.4.3 Dataset

MetaModulus Dataset. In this section, we describe the details of MetaModulus (Lumpe & Stankovic, 2021) dataset, which contains three mechanical properties, *i.e.*, Young’s modulus, Shear modulus, and Poisson’s ratio. This dataset provides a comprehensive mechanical illustration for a metamaterial. Specifically, this dataset contains 16,707 samples originally. To filter out some invalid structures with clustered nodes and non-edge nodes, we conduct two criteria in data preprocessing by following previous works (Zheng et al., 2023; Lumpe & Stankovic, 2021; Chen et al., 2025b): (1) filtering out the structure that contains nodes with less than two edges (dangling nodes). (2) filtering out the structure whose node number is more than 100, since generally a metamaterial structure with too many nodes would be unstable and hard to construct in the real world. Finally, the dataset contains a total of 9871 structures, and we select 8000 for training the VAE (Generator) and others for testing the coverage rate (Cov. R). Figure 11 shows six example structures in this dataset.

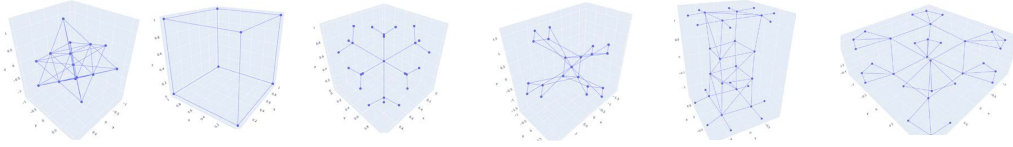


Figure 11: Samples in MetaModulus dataset.

Prompts for Language Guidance. In order to evaluate the language-guidance effectiveness, we introduce 100 design prompts to test if the model can generate effective structures that fit the prompt semantically. These prompt targets on high-level design concepts, containing terms such as, “high-stiffness”, “hard material”, “extremely flexible”, *etc.* Table 3 illustrates 10 example prompts from the prompt dataset. We publish the full design prompts data in our code base.

B.4.4 Implementation Details

METASYMBO contains three agents, among which there are two architectures required to be trained, *i.e.*, the latent generative model in Agent Generator and the property predictor in Agent Supervisor. All training processes are conducted in NVIDIA A100 or H200 GPUs.

Training Details of Latent Generative Model. First, we train the VAE according to the disentangled Evidence Lower Bound (ELBO) loss Eq. (6). After that, if the instantiation is diffusion model (as depicted in Figure 10), we freeze the VAE and train the denoising model according to the score function Eq. (7). In our experiments, we use VAE as the instantiation rather than the diffusion model. In addition, in the training process, we train the VAE for at most 5000 epochs with early stopping trick on the training data without using validation dataset. We select the checkpoint with the lowest training loss.

Training Details of Predictor in Agent Supervisor. After the VAE is trained, we incorporate a predictor head of MLPs in the VAE. Specifically, the encoded latents are fed into the predictor head and output the demanded mechanical properties. In the training process, we freeze parameters in the VAE, and only update the parameters in the Predictor head using Mean Squared Error (MSE) loss between prediction and ground truth. We train it for at most 2000 epochs with early stopping strategy. We randomly select 500 samples from training dataset as validation set, and finally use the checkpoint with the lowest MSE in validation set. Moreover, the predictions and ground truth are max-min normalized for a more stable training.

Training Details of Predictor in Agent Supervisor for Evaluation. In the evaluation process of Section 4.1, we use Supervisor with GPT-4.1 and predictor for scoring. In this phase, the predictor is trained using full dataset, with 500 randomly selected as validation set.

Prompt Details of the LLM in Agent Supervisor for Evaluation. In the evaluation process of Section 4.1, we use Supervisor with GPT-4.1 as evaluator. The overall prompt idea is similar to Figure 9. Differently, we only require it to output scores, without improved properties. Specifically, the prompt in evaluation process is shown in Figure 12.

Table 4: Convergence analysis. Loss change of iterations.

Loss	0	50	100	150	300
$\mathcal{L}_s$	28.6	1.85 (−26.8)	1.21 (−0.64)	0.93 (−0.28)	0.78 (−0.15)
$\mathcal{L}_{p,e}$	147.4	82.2 (−65.2)	78.9 (−3.3)	81.2 (+2.3)	81.9 (+0.7)
$\mathcal{L}_r$	0.0	123.4 (+123.4)	95.8 (−27.6)	85.3 (−10.5)	76.8 (−8.5)
$\mathcal{L}_{prior}$	2.22	2.20 (−0.02)	2.22 (+0.02)	2.24 (+0.02)	2.27 (+0.03)

C More Results

C.1 Convergence Analysis.

Table 4 presents the average loss curve of two randomly selected example with the Mix operator, showing that the $\mathcal{L}_s$ and $\mathcal{L}_{p,e}$ decrease until they converge while other two regularization terms $\mathcal{L}_r$ and $\mathcal{L}_{prior}$ tend to converge to a specific value.

C.2 Validity Experiments for Agent Designer

To further verify Agent 1’s ability to produce reasonable scaffolds, we provide additional examples that can be independently checked.

- **Prompt:** Provide a very stable structure with low stiffness.

```
Node number: 14
Node coordinates (fractional):
(0.0, 0.0, 0.0)
(1.0, 0.0, 0.0)
(0.0, 1.0, 0.0)
(1.0, 1.0, 0.0)
(0.0, 0.0, 1.0)
(1.0, 0.0, 1.0)
(0.0, 1.0, 1.0)
(1.0, 1.0, 1.0)
(0.5, 0.5, 0.0) # face center (z=0)
(0.5, 0.5, 1.0) # face center (z=1)
(0.5, 0.0, 0.5) # face center (y=0)
(0.5, 1.0, 0.5) # face center (y=1)
```

Agent Supervisor Prompt for Evaluation
You are a metamaterial scientist specializing in structural design and mechanical characterization. You are fluent in the geometry and typical property ranges of canonical 3-D architectures such as octet-truss, BCC, SC, Kelvin cell, Diamond, and TPMS families.. Task ----- Given one *design prompt* and a corresponding *metamaterial structure* with its mechanical properties (Young's modulus, Shear modulus, and Poisson's ratio), output: **Score** — a single real number in **[0, 1]** evaluating how well the structure and the provided properties (if have) fulfills the design prompt (0 = poor, 1 = perfect). Input Format ----- Prompt: <free-text design requirement> Structure: ~~~~~ Node number: <N> Node coordinates: (x1, y1, z1) ... (xN, yN, zN) Edges: (i0, j0) ... (iM, jM) ~~~~~ Lattice lengths: [a, b, c] Lattice angles: [α, β, γ] Properties: Young's modulus: [Ex, Ey, Ez] Shear modulus: [Gxy, Gyz, Gzx] Poisson ratio: [vxy, vyx, vxz, vzx, vyz, vzy] Output Format ----- Score: <float in [0,1]> Constraints ----- - Return <i>*only*</i> the fields specified above, in the exact order and layout. - Provide no additional commentary, explanations, or reasoning steps. - For mechanical properties, their value has scales: Ex, Ey, Ez in [0, 1e-2]; Gxy, Gyz, Gzx in [0, 1e-2]; vxy, vyx, vxz, vzx, vyz, vzy in [-20, +20].

Figure 12: The prompt in Agent Supervisor for evaluation.

```
(0.0, 0.5, 0.5) # face center (x=0)
(1.0, 0.5, 0.5) # face center (x=1)
```

Edges:

```
(8, 0)
(8, 1)
(8, 2)
(8, 3)
(9, 4)
(9, 5)
(9, 6)
(9, 7)
(10, 0)
(10, 1)
(10, 4)
(10, 5)
(11, 2)
(11, 3)
(11, 6)
(11, 7)
(12, 0)
(12, 2)
(12, 4)
(12, 6)
(13, 1)
(13, 3)
(13, 5)
(13, 7)
```

Structural rationale:

High Stability: Triangle crossed struts increase stability. Octet structure is stable.

Low Stiffness: Minimal edge connectivity along principal directions reduces stiffness, providing flexibility and deformation capacity under loading.

- **Prompt:** Provide a BCC structure.

Node number: 9

coordinates:

```
(0,0,0)
(1,0,0)
(1,1,0)
(0,1,0)
(0,0,1)
(1,0,1)
(1,1,1)
(0,1,1)
(0.5,0.5,0.5)
```

Edges:

```
(0,8)
(1,8)
(2,8)
(3,8)
(4,8)
(5,8)
(6,8)
(7,8)
```

According to the results, we can visualize the scaffold examples as in Figures 13.

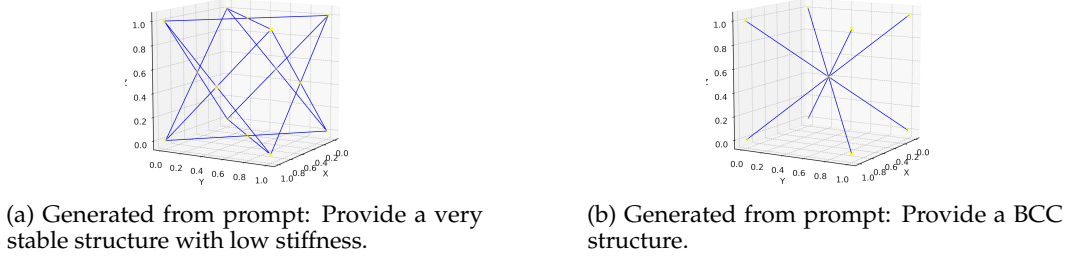


Figure 13: Visualization of two generated unit cells produced by Agent Designer.

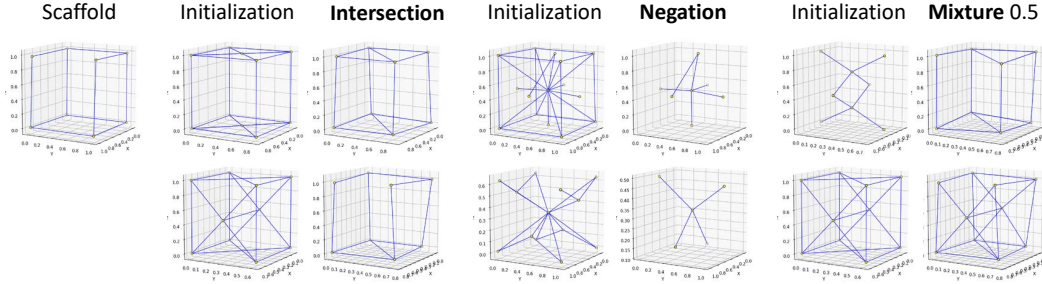


Figure 14: Qualitative results of Intersection, Negation, and Mixture symbolic operators.

C.3 Additional Qualitative Analysis of Intersection and Negation.

We provide additional qualitative results for the Intersection and Negation operators in Figure 14. Unlike Union and Mix, these two operators are not intended to produce direct geometric merging or interpolation. Instead, they operate at the semantic level in latent space. Intersection emphasizes the common semantic components shared by the initialization and the scaffold, encouraging the generated structure to retain only their mutually consistent attributes. Negation, by contrast, suppresses scaffold-related latent factors, steering the generation away from the emphasized semantics of the scaffold. As a result, their outputs should not be interpreted as literal geometric intersection or subtraction in Euclidean space, but rather as semantic transformations induced in the disentangled latent space. The qualitative results are consistent with this design: both operators produce meaningful structural changes, while exhibiting less direct controllability than Union and Mix, which is why we focus on the latter two in the main text.

C.4 More Analysis for Property Predictor in Agent Supervisor

Figure 15 shows the fitting performance of the proposed disentangled predictor on the test set regarding three mechanical properties (*i.e.*, Young’s modulus, Shear modulus, and Poisson’s ratio). It can be seen that all fitting $R^2 > 0.8$, demonstrating effective fitting performance. Mover, Table 5 compared our proposals with several existing works, including invariant model SphereNet (Liu et al., 2022), Equivariant model (Liao & Smidt, 2023) and ViSNet (Wang et al., 2024), and valina VAE, demonstrating the superiority of disentangled semantic latent in effectively capturing mechanical proeptrties.

C.5 More Case Studeis

In this section, we show more case studies given a design concept. Figures 16 and 17 show two other case studies, which demonstrate the effectiveness of multi-agent collaborations. Specifically, Case study 2 (Figure 16) proposes a simple scaffold that has extreme flexibility at first, and combines it with a possible structure that contains the most suitable mechanical properties to generate final results. In addition, Case study 3 (Figure 17) shows that the

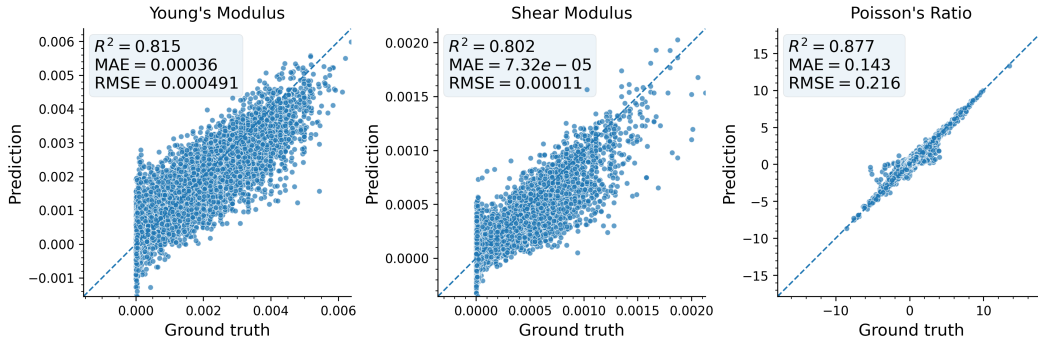


Figure 15: Prediction results of the proposed disentangled VAE on three properties. All $R^2 > 0.8$ demonstrates the strong prediction results on the three mechanical properties.

Table 5: Performance comparison of different models on predicting mechanical properties (MAE). Show the superior performance of the proposed framework.

Models	Young's Modulus	Shear Modulus	Poisson's Ratio
Vanilla VAE	$6.3e^{-4}$	$1.4e^{-4}$	0.39
SphereNet (Liu et al., 2022)	$4.7e^{-4}$	$1.0e^{-4}$	0.35
Equiformer (Liao & Smidt, 2023)	$6.6e^{-4}$	$2.2e^{-4}$	0.36
ViSNet (Wang et al., 2024)	$6.2e^{-4}$	$6.3e^{-2}$	0.37
MetaSymbO (disentangled VAE)	$3.5e^{-4}$	$7.3e^{-5}$	0.14

evaluation score increases with multi-agent collaboration iterations. Finally, it generates a complicated but suitable structure with high density, which implies high rigidity.

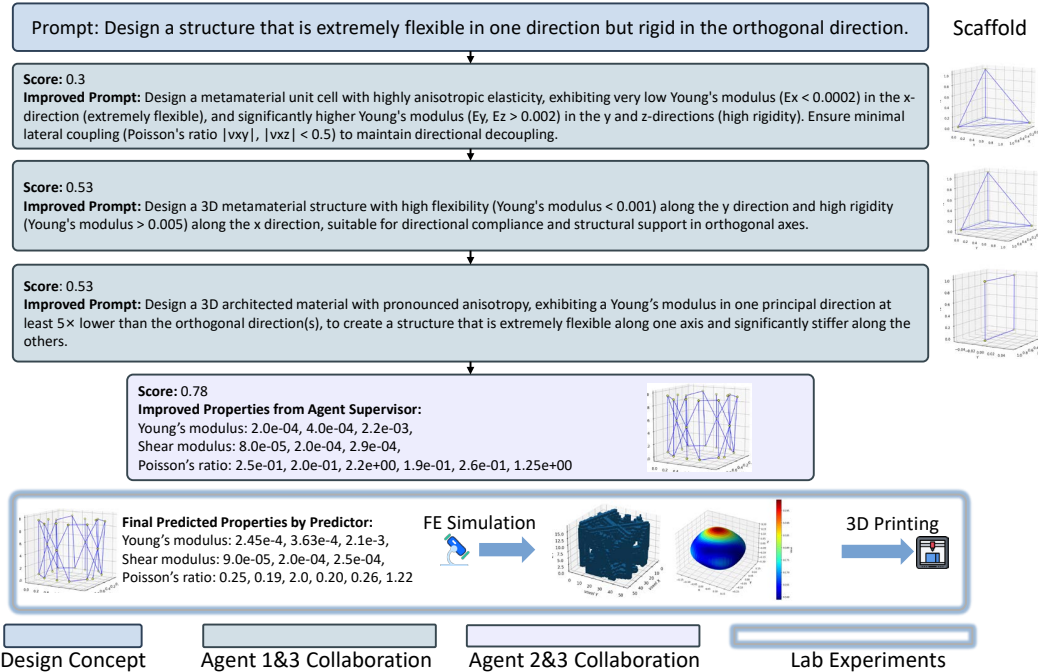


Figure 16: Second case study.

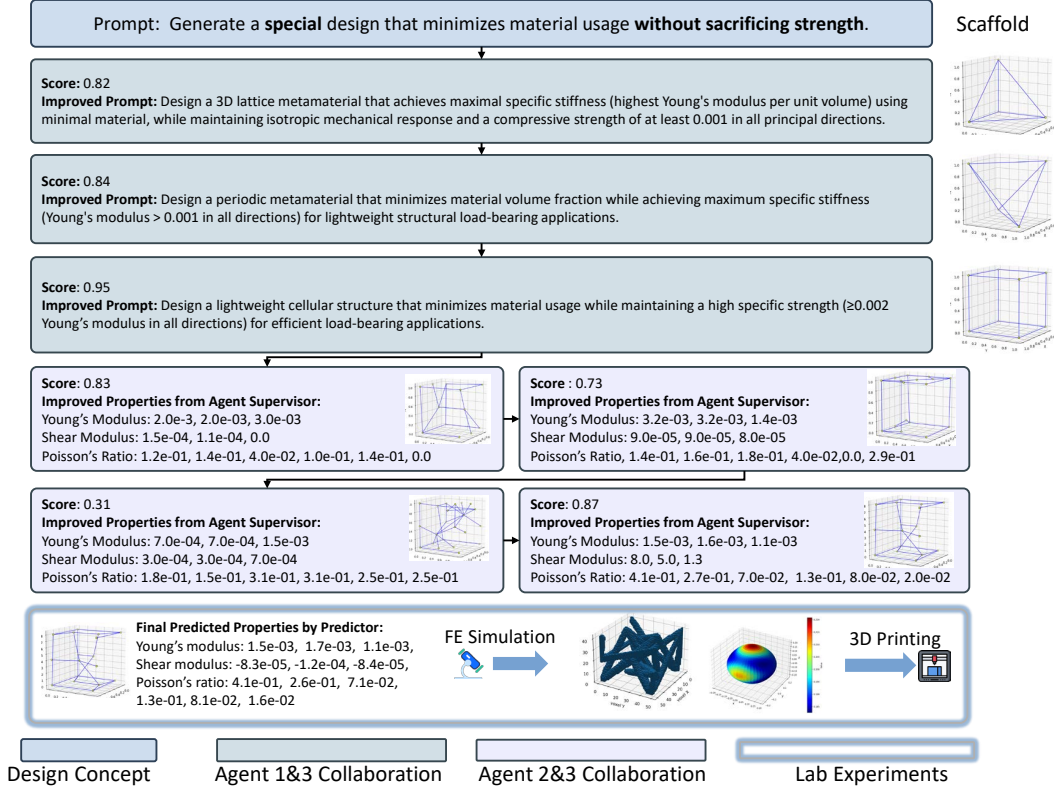


Figure 17: Third case study.

C.6 Wet Lab Results

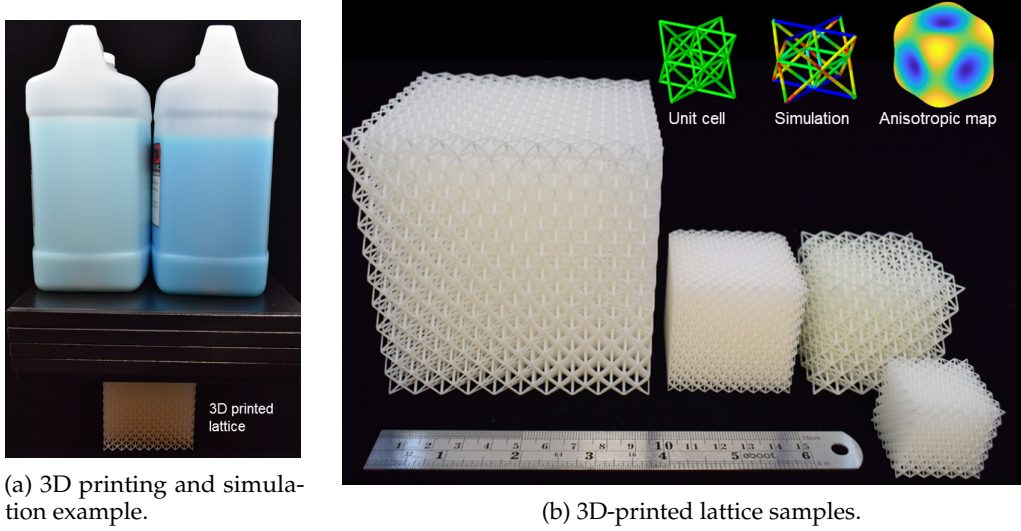
For wet-lab validation, Figures 18a and 18b present representative 3D-printed lattice specimens obtained by periodically tiling the generated truss-like unit cells into macroscopic samples, consistent with the lattice-based metamaterial formulation of the framework.

Figure 18a shows a small proof-of-concept print placed next to common laboratory containers, providing an intuitive reference for the physical scale of the fabricated structure and the fine resolution of its strut network. Figure 18b further shows several larger printed samples with different overall sizes and shapes, together with the corresponding unit-cell, simulation, and anisotropy visualizations, illustrating the full pipeline from digital design to physical realization. The preserved open-cell topology, regular repetition of unit cells, and visually clean node–edge connectivity indicate that the generated structures are practically printable, supporting the use of 3D printing as a wet-lab validation step for the final designs in our case study.

D Theoretical Analysis

D.1 Latent Generation Models

Auto-encoder (AE) is widely used in the material discovery domain to compress discrete material data to a continuous latent space for downstream tasks (especially inverse material design) (Zeng et al., 2025; Hanakata et al., 2020). VAEs (Kingma & Welling, 2014) introduce a Gaussian prior and derive evidence lower bound optimization (ELBO). More recently, latent diffusion models (LDMs) are extensively explored due to its strong ability in reconstructing high-fidelity data (Rombach et al., 2022; Podell et al., 2023; Fu et al., 2024). Both VAEs and



(a) 3D printing and simulation example.

(b) 3D-printed lattice samples.

Figure 18: Wet-lab validation via 3D printing. Left: a proof-of-concept printed lattice with simulation context. Right: representative 3D-printed lattice samples at different scales, demonstrating the manufacturability of the generated periodic structures.

LDMs typically impose a multivariate Gaussian prior on the latent space, thereby enabling the application of symbolic logic operators directly within the latent Gaussian manifold.

Therefore, Agent Generator is instantiated as an AE-based latent generation model in this work, and we implement the basic version of VAEs and LDMs for experiments.

Formally, given a metamaterial $M = (\mathbf{L}, \mathcal{U})$, Both VAEs and LDMs operate by encoding M into a continuous latent variable $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^d$ via a stochastic encoder: $q_\phi(\mathbf{z} | M)$, and reconstruct the input through a decoder: $p_\theta(M | \mathbf{z})$. Similar for VAEs and LDMs, a standard Gaussian prior is imposed over the latent space: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I})$. The difference lies in the objectives. Specifically, VAEs conduct ELBO for optimization:

$$\mathcal{L}_{\text{VAE}}(\phi, \theta) = \mathbb{E}_{q_\phi(\mathbf{z}|M)} [\log p_\theta(M | \mathbf{z})] - \text{KL}(q_\phi(\mathbf{z} | M) \| p(\mathbf{z})), \quad (19)$$

while in LDMs, latent variables $\mathbf{z}_0$ are further corrupted over T steps via a diffusion process and denoising process:

$$\begin{aligned} q(\mathbf{z}_t | \mathbf{z}_0) &= \mathcal{N}(\mathbf{z}_t | \sqrt{\bar{\alpha}_t} \mathbf{z}_0, (1 - \bar{\alpha}_t) \mathbf{I}), \\ p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t) &= \mathcal{N}(\mathbf{z}_{t-1} | \boldsymbol{\mu}_\theta(\mathbf{z}_t, t), \boldsymbol{\Sigma}_\theta(t)), \end{aligned} \quad (20)$$

where $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, and $\{\alpha_t\}$ is a predefined noise schedule. The training objective minimizes the expected noise prediction to learn the parameters θ in denoising model $\mathcal{L}_{\text{LDM}}(\theta) = \mathbb{E}_{\mathbf{z}_0, \epsilon, t} \|\epsilon - \epsilon_\theta(\mathbf{z}_t, t)\|^2$, where $\mathbf{z}_t = \sqrt{\bar{\alpha}_t} \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$, and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is standard Gaussian noise.

D.2 Derivation of Symbolic Operators in Gaussian

D.2.1 Derivation of Mix Operator (Eq. (11))

Given two diagonal-Gaussian latent posteriors $q_M(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_M, \text{diag } \sigma_M^2)$ and $q_{M'}(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_{M'}, \text{diag } \sigma_{M'}^2)$, their convex combination is

$$p_{\text{mix}}(\mathbf{z}) = (1 - \lambda_{\text{mix}}) q_M(\mathbf{z}) + \lambda_{\text{mix}} q_{M'}(\mathbf{z}), \quad \lambda_{\text{mix}} \in [0, 1]. \quad (21)$$

$$\mathbb{E}_{p_{\text{mix}}}[\mathbf{z}] = (1 - \lambda_{\text{mix}}) \boldsymbol{\mu}_M + \lambda_{\text{mix}} \boldsymbol{\mu}_{M'} = \boldsymbol{\mu}_{\text{mix}}.$$

Let $\Sigma_M = \text{diag } \sigma_M^2$ and $\Sigma_{M'} = \text{diag } \sigma_{M'}^2$. Then

$$\Sigma_{\text{mix}} = (1 - \lambda_{\text{mix}})\Sigma_M + \lambda_{\text{mix}}\Sigma_{M'} + \lambda_{\text{mix}}(1 - \lambda_{\text{mix}})(\mu_M - \mu_{M'})(\mu_M - \mu_{M'})^\top. \quad (22)$$

To maintain diagonal structure and avoid the expensive cross term, we drop the outer-product term and interpolate standard deviations:

$$\sigma_{\text{mix},k} \approx (1 - \lambda_{\text{mix}})\sigma_{M,k} + \lambda_{\text{mix}}\sigma_{M',k}.$$

With σ_{mix} defined above and a small constant ε for numerical stability,

$$p_{\text{mix}}(\mathbf{z} \mid \lambda_{\text{mix}}) = \mathcal{N}(\mathbf{z}; (1 - \lambda_{\text{mix}})\mu_M + \lambda_{\text{mix}}\mu_{M'}, \text{diag}((1 - \lambda_{\text{mix}})\sigma_M + \lambda_{\text{mix}}\sigma_{M'})^2 + \varepsilon) \quad (23)$$

Remark. Equation 23 is a *heuristic single-Gaussian approximation*: it preserves the exact mean but underestimates the true covariance by omitting the cross term in 22. This trade-off yields a numerically stable, fully differentiable latent representation while retaining controllable guidance via λ_{mix} .

D.2.2 Derivation of Intersection Operator (Eq. (13))

Recall the formula of PoE (Eq. (12)) and the Gaussian distribution of p_M and $p_{M'}$, we have the following derivation by following (Kant et al., 2024; Hinton, 2002).

$$\begin{aligned} p_{\text{int}}(\mathbf{z}) &\propto p_M(\mathbf{z}) p_{M'}(\mathbf{z}) \\ &= \underbrace{\frac{1}{(2\pi)^{d/2} |\Sigma_M|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{z} - \mu_M)^\top \Sigma_M^{-1}(\mathbf{z} - \mu_M)\right]}_{p_M(\mathbf{z})} \\ &\quad \underbrace{\frac{1}{(2\pi)^{d/2} |\Sigma_{M'}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{z} - \mu_{M'})^\top \Sigma_{M'}^{-1}(\mathbf{z} - \mu_{M'})\right]}_{p_{M'}(\mathbf{z})} \\ &\propto \exp\left[-\frac{1}{2}\left(\mathbf{z}^\top (\Sigma_M^{-1} + \Sigma_{M'}^{-1})\mathbf{z} - 2\mathbf{z}^\top (\Sigma_M^{-1}\mu_M + \Sigma_{M'}^{-1}\mu_{M'})\right)\right]. \end{aligned}$$

Let the *precision* matrix be $\Lambda_{\text{int}} = \Sigma_M^{-1} + \Sigma_{M'}^{-1}$ and define $\eta_{\text{int}} = \Sigma_M^{-1}\mu_M + \Sigma_{M'}^{-1}\mu_{M'}$. Completing the square gives

$$p_{\text{int}}(\mathbf{z}) \propto \exp\left[-\frac{1}{2}(\mathbf{z} - \mu_{\text{int}})^\top \Lambda_{\text{int}}(\mathbf{z} - \mu_{\text{int}})\right], \quad \mu_{\text{int}} = \Lambda_{\text{int}}^{-1}\eta_{\text{int}}.$$

Since $\Sigma_{\text{int}} = \Lambda_{\text{int}}^{-1}$, we arrive at

$$p_{\text{int}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mu_{\text{int}}, \Sigma_{\text{int}}), \quad \Sigma_{\text{int}} = (\Sigma_M^{-1} + \Sigma_{M'}^{-1})^{-1}, \quad \mu_{\text{int}} = \Sigma_{\text{int}}(\Sigma_M^{-1}\mu_M + \Sigma_{M'}^{-1}\mu_{M'}).$$

Remark. The intersection (PoE) weights each mean by its precision, yielding a sharper Gaussian concentrated where both experts agree.

D.2.3 Derivation of Negation Operator (Eq. (15))

Similar to the derivation of PoE (Kant et al., 2024; Hinton, 2002), we can derive the following for the negation operator.

Remember that the original and scaffold latent distributions are two multivariate Gaussians

$$p_M(\mathbf{z}) = \mathcal{N}(\mu_M, \Sigma_M), \quad p_{M'}(\mathbf{z}) = \mathcal{N}(\mu_{M'}, \Sigma_{M'}).$$

We suppress the density peaks of M' inside M by defining

$$p_{\text{neg}}(\mathbf{z}) \propto \frac{p_M(\mathbf{z})^\alpha}{p_{M'}(\mathbf{z})^\beta}, \quad \alpha, \beta > 0.$$

Because the log-density of any Gaussian is quadratic, we write

$$\begin{aligned} \log p_{\text{neg}}(\mathbf{z}) &= \alpha \log p_M(\mathbf{z}) - \beta \log p_{M'}(\mathbf{z}) + \text{const} \\ &= -\frac{1}{2} \mathbf{z}^\top (\alpha \Sigma_M^{-1} - \beta \Sigma_{M'}^{-1}) \mathbf{z} + \mathbf{z}^\top (\alpha \Sigma_M^{-1} \boldsymbol{\mu}_M - \beta \Sigma_{M'}^{-1} \boldsymbol{\mu}_{M'}) + \text{const}. \end{aligned}$$

After that, to ensure the Gaussian, we conduct the moment-matched Gaussian approximation. Specifically, treat the quadratic form above as the (unnormalised) log of a new Gaussian. Define the *negation precision*

$$\Lambda_{\text{neg}} \triangleq \alpha \Sigma_M^{-1} - \beta \Sigma_{M'}^{-1}, \quad (\Lambda_{\text{neg}} \succ 0 \text{ required}),$$

and invert it to obtain the covariance $\Sigma_{\text{neg}} = \Lambda_{\text{neg}}^{-1}$.

Multiplying the linear term by Σ_{neg} gives the mean:

$$\boldsymbol{\mu}_{\text{neg}} = \Sigma_{\text{neg}} (\alpha \Sigma_M^{-1} \boldsymbol{\mu}_M - \beta \Sigma_{M'}^{-1} \boldsymbol{\mu}_{M'}).$$

Finally, we obtain the approximated Gaussian as

$$p_{\text{neg}}(\mathbf{z}) \approx \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\text{neg}}, \Sigma_{\text{neg}}), \quad \Sigma_{\text{neg}}^{-1} = \alpha \Sigma_M^{-1} - \beta \Sigma_{M'}^{-1}.$$

The approximation is valid only when $\alpha \Sigma_M^{-1} \succ \beta \Sigma_{M'}^{-1}$, ensuring Λ_{neg} (and hence Σ_{neg}) is positive definite.

Remark. Choosing $\alpha > \beta$ or scaling $\Sigma_{M'}$ slightly upward guarantees the precision matrix remains positive definite, making the negation operator a well-defined Gaussian.