LEARNING LATENT DESIGN SPACES FOR EFFICIENT OPTIMIZATION OF PIXELATED ELECTROMAGNETIC STRUCTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixelated electromagnetic (EM) structures offer substantial design flexibility, but their high-dimensional, combinatorial design spaces and costly full-wave simulations make target-driven optimization challenging. We investigate whether unsupervised representation learning can provide compact latent design spaces for efficient optimization of these structures. Engineering knowledge is encoded as geometric rules that generate large collections of layouts without EM response labels. We compare variational autoencoders, adversarial autoencoders, and an encoder-equipped Wasserstein generative model (WGAN-encoder) for learning low-dimensional representations of these geometric priors. Bayesian optimization then searches the learned spaces, using full-wave simulations to evaluate decoded candidates. We evaluate this pipeline on reflection metasurfaces (reflectarrays) and on-chip transformers, with four- and six-dimensional latent spaces, respectively. All three model families support effective downstream optimization, although convergence and final performance vary with the model and its regularization. A complete 16-cell reflection-phase library is obtained with 63 full-wave unit-cell evaluations and validated in a fabricated reflectarray, while transformer optimization improves the objective across all 16 target specifications. These results support learning from inexpensive geometric data as a practical approach to simulation-efficient optimization of pixelated EM structures across different representation models and device types.

1 INTRODUCTION

An electromagnetic (EM) structure is a physical arrangement of materials that controls how electromagnetic energy propagates, reflects, or couples between conductors. Examples range from patterned surfaces that shape an antenna beam to metal windings that couple signals inside a chip. Changing where metal is placed changes current paths and electromagnetic fields, and hence the device's response. Inverse design seeks a geometry that produces a desired response, evaluated through full-wave simulations that numerically solve Maxwell's equations.

Pixelated representations make this problem accessible to learning algorithms. A layout becomes a binary image, or a stack of images for multiple material layers, in which each entry specifies the presence or absence of metal. Such representations have enabled learning-based synthesis of multi-port radio-frequency (RF) devices (Karahan et al., 2024) and diffusion-based EM generation (Guo et al., 2025). Each pixel describes a material choice in a physical device: changing a few pixels can connect conducting regions or close a gap, altering how currents flow. A useful design representation should therefore capture the geometric relationships that govern device behavior.

The attraction of pixelation is its freedom to express geometries beyond an individual hand-designed template. Its difficulty is the cost of exploring that freedom. An unconstrained layout with D binary choices contains 2^D configurations. Geometric constraints reduce this set but do not by themselves provide continuous coordinates for search. Evaluating a candidate generally requires an expensive EM simulation. Learning a response predictor can accelerate evaluation, but collecting its geometry–response training pairs also requires substantial simulation effort.

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

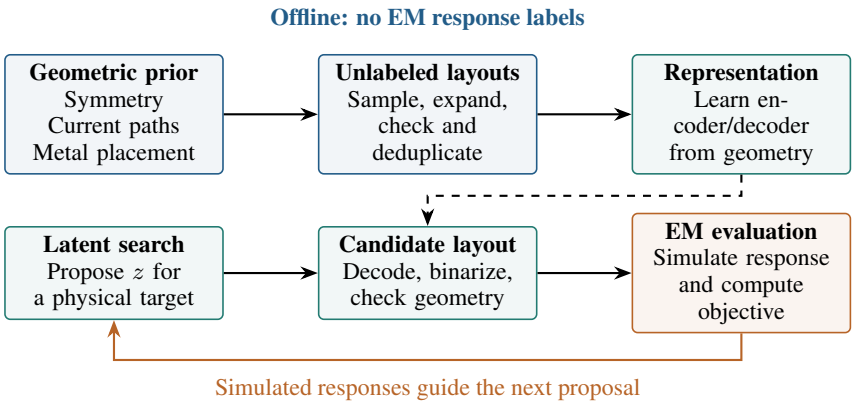


Figure 1: From engineering knowledge to simulation-guided design. Offline, geometric rules generate unlabeled layouts for representation learning. Online, the decoder converts an optimizer's latent proposal into a candidate layout, which is processed and evaluated by the EM solver. The simulated response guides the next proposal. Training the representation requires geometry alone; the physical target enters only during optimization.

Engineering knowledge offers a different starting point. Physical intuition and design experience suggest useful geometric features: symmetries suited to a device's operation, connected paths for current flow, and separation between neighboring metal patterns. Such knowledge can be encoded as rules that generate many structurally plausible layouts without simulating each one. These rules provide a prior over candidate structures; simulation is then used during optimization to determine which candidates meet a quantitative performance target.

We use this prior to learn the space in which optimization takes place. First, a procedural generator samples a large geometry-only dataset from the rules. A representation model then learns a compact latent space and a decoder that maps latent coordinates to pixel layouts. Finally, an optimizer searches these coordinates, using EM simulations to evaluate decoded candidates. Figure 1 summarizes the sequence: *geometric prior, unlabeled layouts, learned representation, and simulation-guided search*. The learning stage absorbs recurring geometric structure, while physical evaluations are concentrated on the subsequent design task.

Representation learning aims to convert rule-defined geometry distribution into coordinates for continuous search. The resulting space should support both structurally valid generation and target-driven optimization. We examine reconstruction, generation validity, and downstream search to assess how different representation models support physical design.

We investigate reflection metasurfaces and on-chip transformers as two applications of this workflow, which use different geometric priors within the same pipeline. Experiments compare representation quality in both applications, generation validity for transformers, and downstream BO across model families for reflectarray unit cells. A fabricated reflectarray provides additional physical validation. Our contributions are the common engineering pipeline for pixelated EM structures, with comparison of latent design spaces from different models, and validation through target-driven synthesis and the fabricated reflectarray.

2 Related Work

Learning for EM inverse design. Pixelated representations support learning-based radio-frequency (RF) device design. Karahan et al. (2024) train a forward model that predicts electromagnetic responses from layouts, and Dall-EM generates layouts conditioned on target scattering parameters using diffusion (Guo et al., 2025). These approaches use geometry–response training pairs, whose collection requires physical evaluation. Our representation-learning stage uses layouts generated from geometric priors, with response data collected during subsequent optimization.

Latent-space optimization has also been applied to physical structures. Kudyshev et al. (2021) optimize 64×64 binary photonic layouts using a conditional AAE and differential evolution, with topology-optimized training designs and a learned performance predictor. Zhu et al. (2022) combine an AAE and BO for 36-layer binary material sequences. Tian et al. (2024) pretrain a VAE on unlabeled geometry and refine it using objective feedback; evaluated candidates are selected by matching decoded images to an existing geometry library. In our workflow, the decoded material pattern itself becomes the candidate after binarization and geometric processing. Our focus is learning from inexpensive geometric priors and optimizing newly decoded pixel layouts through full-wave simulation.

Representation learning for optimization. VAEs (Kingma & Welling, 2013) and AAEs (Makhzani et al., 2015) combine reconstruction with latent regularization; Wasserstein GANs (Arjovsky et al., 2017; Gulrajani et al., 2017) use an image-space critic, and Wasserstein autoencoders (Tolstikhin et al., 2018) combine reconstruction with latent distribution matching. Weighted retraining (Tripp et al., 2020) and LOL-BO (Maus et al., 2022) further adapt representations using objective feedback.

3 From geometric priors to latent optimization

The workflow has three stages: engineering knowledge defines a distribution of candidate layouts, representation learning converts it into a latent design space, and simulation-guided optimization searches that space for a physical target. We instantiate these stages separately for reflectarray unit cells and on-chip transformers, beginning with what is designed and how a design is scored.

3.1 Design variables and physical targets

A reflectarray is a patterned surface that redirects a wave from a feed antenna into a desired beam. It consists of an array of patterned unit cells. Each cell changes the phase of the reflected wave, meaning its timing relative to the incident wave. With appropriate phases across the surface, the reflected waves reinforce one another in the desired direction. The unit-cell task is therefore to obtain a set of cells that together supply the full range of reflection phases. An on-chip transformer is a component in radio-frequency integrated circuits that couples signals through two metal windings. Here, "transformer" refers to this physical device rather than the neural-network architecture. Its design targets the inductance of each winding and their magnetic coupling.

Both devices are described by the same design variable, $x \in \{0,1\}^{C\times H\times W}$, with C patterned layers and binary entries denoting the presence or absence of metal. Engineering priors are encoded by geometric rules $\mathcal{R}$, such as symmetry, material placement, and permitted contacts, which define a candidate set $\mathcal{X}_{\mathcal{R}}$. A full-wave simulator $s(x;c)$ evaluates the response under configuration c, including the substrate, ground plane, excitation, and frequency. Given a target t, optimization seeks

$$\min_{x\in\mathcal{X}_{\mathcal{R}}} f_t(x;c), \qquad f_t(x;c) = \ell(s(x;c),t), \tag{1}$$

within a finite simulation budget. Geometric rules define the candidate family; the simulated response determines whether a candidate achieves the target. We next describe how the prescribed rules are turned into a searchable space.

3.2 Constructing a geometry-only prior

For the reflectarray, square symmetry lets a cell treat the two orthogonal polarizations of the incident wave alike. An empty outer border keeps metal in adjacent cells apart, and excluding corner-only contacts avoids zero-width metallic junctions. These design choices narrow the search to a physically motivated family that still allows many arrangements of metal inside each cell. Metal is grown in a symmetry-reduced region from fixed pixels and then expanded by the rotations and reflections of a square (the group D_4) into a complete layout. Repeating this with random choices and rejecting rule violations yields tens of thousands of layouts without EM response labels; Figure 2 (left) shows examples.

The transformer prior (Figure 2, right) instead encodes winding geometry. Its generator combines conventional circular, rectangular, and octagonal contours with node-controlled variations and local

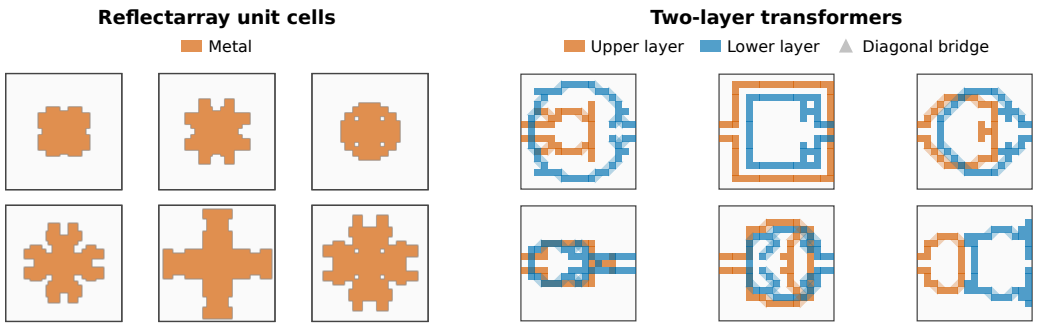


Figure 2: Samples from the two geometry-only design priors; none requires an EM simulation. Left: reflectarray unit cells produced by symmetric motif growth, each with D_4 symmetry and an empty boundary ring. Right: transformer layouts with conventional and node-controlled contours; the two colors denote the two metal layers, and light triangles give finite area to diagonal contacts. Orange denotes metal in both applications.

horizontal, vertical, diagonal, L-shaped, and T-shaped motifs. Each layout contains two 17×17 metal layers. Acceptance requires fixed ports, horizontal reflection symmetry, a path joining the ports, and no floating metal. Permitted diagonal contacts receive finite-area bridges before simulation.

Formally, let $A_{\mathcal{R}}(u)$ be a procedural sampler, where u collects random geometric choices. Conditioned on geometric acceptance, the training distribution is

$$\mathcal{D}_{\text{geom}} = \{x_i\}_{i=1}^{N}, \qquad x_i \sim p_{\mathcal{R}}, \qquad p_{\mathcal{R}} = \text{Law}(A_{\mathcal{R}}(u) \mid \text{acceptance}). \tag{2}$$

The rules, sampling procedure, and deduplication policy together determine this prior. No response $s(x_i; c)$ is needed to obtain a training example. Appendix A describes the geometric sampling procedures and datasets.

The sampled layouts capture the geometric prior in discrete form. The next stage learns a representation of this distribution in which a low-dimensional continuous code controls the arrangement of many pixels.

3.3 Learning the latent design space

An encoder E_ϕ maps a layout to $z \in \mathbb{R}^d$, and a decoder or generator G_θ maps a code to a continuous-valued layout. We compare three representation families. Their objectives differ in how they preserve input geometry and organize latent codes; the downstream experiment evaluates how those choices affect optimization.

Variational autoencoder. The VAE learns a posterior $q_\phi(z \mid x)$ and a decoder likelihood. Reconstruction encourages recovery of the input layout, while a KL penalty pulls each encoded distribution toward the sampling prior (Appendix B).

Adversarial autoencoder. The AAE also reconstructs its input but regularizes the aggregate distribution of codes across the dataset, rather than each encoded distribution, toward $p(z)$. A latent discriminator D_ψ distinguishes prior samples from encoded layouts, and the encoder learns to make them indistinguishable. For pixelated transformer design, we compare Gaussian and uniform priors in a six-dimensional latent space (Appendix B).

WGAN-encoder. The third family starts from a WGAN-GP generator rather than an autoencoder and adds an encoder trained through latent cycle consistency and maximum mean discrepancy (MMD). With image-space critic C_ω, the joint generator/encoder objective is

$$\begin{aligned}\mathcal{L}_{G,E} = & -\mathbb{E}_z C_\omega(G_\theta(z)) + \frac{\lambda_{\text{cyc}}}{d}\mathbb{E}_z\|E_\phi(G_\theta(z)) - z\|_2^2 \\ & + \lambda_{\text{MMD}}\,\text{MMD}_k^2\left(\text{Law}(E_\phi(G_\theta(z))), \text{Law}(E_\phi(x))\right).\end{aligned} \tag{3}$$

The critic distinguishes generated layouts from training layouts and uses the gradient penalty of Gulrajani et al. (2017). The cycle term encourages recovery of the code that generated a layout. MMD encourages agreement between the distributions of encoded real and generated layouts. The encoder thus influences how the generator organizes its codes, although online candidate generation requires only the trained generator.

These losses train different capabilities. VAE and AAE objectives explicitly reconstruct real layouts; Equation 3 recovers generated codes and matches distributions, with no explicit real-layout reconstruction term. Consequently, reconstruction alone is insufficient to compare their suitability for design. The experiments evaluate both reconstruction and downstream search.

3.4 Decoding and simulation-guided search

After training, the latent representation is fixed. Let $P_{\mathcal{R}}$ denote the geometry conversion applied to decoder output, including thresholding, symmetry operations, and contact processing. Candidates that fail the implemented checks are rejected or repaired before evaluation. Optimization then becomes

$$\min_{z \in \mathcal{Z}} \ell(s(P_{\mathcal{R}}(G_\theta(z)); c), t). \tag{4}$$

This replaces the combinatorial search of Equation 1 with a continuous, low-dimensional one, while every evaluated candidate remains a concrete layout.

We use Bayesian optimization (BO) to search the latent space. BO fits Gaussian process (GP) models to evaluated codes and their losses or response metrics, selects a new code with an improvement-based acquisition function, and updates the models after simulation. Reflectarray library synthesis uses a GP for the scalar phase-error objective and LogEI (Ament et al., 2023); the separate model-comparison experiment uses expected improvement (EI). Transformer synthesis uses five GPs for the two inductances, two quality factors, and coupling magnitude, with Monte Carlo qLogEI applied to their joint objective. One candidate is simulated at a time, and the latent decoder remains unchanged as observations accumulate.

4 Evaluation on reflectarrays and transformers

The experiments evaluate the workflow from representation learning to physical design. We first compare reconstruction in both applications and generation validity for transformers (Section 4.2). A five-seed reflectarray experiment then compares downstream BO across representation models (Section 4.3). Finally, we evaluate reflection-phase library synthesis, fabricated-reflectarray performance, and transformer target matching. Together, these studies assess how the learned spaces support different stages of EM design.

4.1 Design tasks and evaluation

Reflection targets. Reflection-phase library synthesis uses a grounded Rogers 4003C unit cell with 10 mm periodicity and 1.524 mm substrate thickness. The objective averages circular phase error, normalized by 180°, at 10, 9.5, and 10.5 GHz with weights $(0.6, 0.2, 0.2)$. Sixteen targets spaced by 22.5° cover the phase cycle; optimization stops when center-frequency error falls below 5°. Appendix D gives the formal objective. The library run starts with 16 candidates and reuses responses across targets. A separate five-seed experiment compares search performance across models (Table 3).

Transformer targets. The transformer study comprises 16 targets specifying primary inductance L_1, secondary inductance L_2, and coupling magnitude $|k|$. Inductances span 25–120 pH, and coupling magnitudes span 0.20–0.60. Inductance characterizes magnetic energy storage for a given current, while coupling measures how strongly the windings interact. Quality factors Q_1, Q_2, which increase as loss decreases, enter as secondary preferences. Metrics are evaluated at 10 GHz (Appendix E).

Table 1: Reflectarray representation quality on layouts. All models use four latent dimensions and 16×16 maps. Arrows indicate the preferred direction; dashes indicate losses not used by the corresponding model.

Metric	VAE	AAE	WGAN-encoder
BCE ↓	0.2388	**0.2297**	–
Mean IoU ↑	0.7258	**0.7305**	0.4573
Mean F1 ↑	0.8366	**0.8397**	0.6147
Latent-cycle MSE ↓	–	–	0.1073
Encoded-distribution MMD^2 ↓	–	–	0.0110

Table 2: Transformer representation quality. Reconstruction and generation measure different uses of the learned decoder. Arrows indicate the preferred direction. All models use six latent dimensions.

Metric	VAE, Gaussian	AAE, Gaussian	AAE, uniform
BCE ↓	0.1465	**0.1332**	0.1343
Mean IoU ↑	0.7350	**0.7650**	0.7629
Mean F1 ↑	0.8472	**0.8668**	0.8655
Raw reconstruction validity ↑	95.30%	98.18%	**98.88%**
Prior-sample validity ↑	78.32%	95.90%	**98.44%**

4.2 Reconstruction fidelity and valid generation

Table 1 compares representation quality on the same 1,000 layouts sampled from the shared 20,000-layout training dataset. Mean intersection over union (IoU) and mean F1 measure binary reconstruction overlap, averaged across individual layouts. The AAE achieves the highest overlap and lowest binary cross-entropy (BCE). WGAN-encoder is trained through latent-cycle consistency and distribution matching, without a paired reconstruction objective; we therefore also report its latent-cycle mean squared error (MSE) and encoded-distribution MMD^2. Appendix B.1 specifies the evaluation procedure.

For transformers, IoU and F1 are computed from pooled pixel counts for each layer and then averaged over the two layers. We also compare reconstruction with valid generation (Table 2). Reconstruction validity measures the fraction of reconstructed layouts satisfying the geometric rules; prior-sample validity applies the same checks to layouts decoded from codes drawn from the latent prior. The latter measures valid generation beyond encoded training examples.

The two AAEs have similar overlap scores, with prior-sample validity of 95.90% for the Gaussian prior and 98.44% for the uniform prior. The VAE achieves a mean reconstruction F1 of 0.8472 and prior-sample validity of 78.32%. These measures capture complementary properties: reconstruction assesses encoded layouts, while prior sampling assesses generation at new codes. For the uniform-prior AAE, 98.88% of 10,000 held-out reconstructions are valid before minimal geometric projection and 99.46% afterward.

Figure 3 illustrates how latent coordinates control geometry. Each row varies one latent variable while the others remain at zero. The decoded layouts show coordinated changes in size, arm shape, winding contour, or layer overlap. These examples illustrate how a small number of continuous variables can control changes across a pixelated layout.

4.3 Comparison of representation models

Figure 4 compares VAE, AAE, and three Wasserstein variants. WG00 omits the encoder regularizers, while WG11 and WG12 use $(\lambda_{\text{MMD}}, \lambda_{\text{cyc}}) = (1, 1)$ and $(1, 2)$. For WG00, Equation 3 gives the encoder no training signal; the generator is still trained by the image-space critic. The comparison therefore examines the effect of latent regularization as well as differences between model families.

The trajectories show improvements during search for all five models. Final mean loss decreases from 0.290 for WG00 to 0.249 for WG11 and 0.157 for WG12. The corresponding mean phase errors are $42.65°$, $38.69°$, and $10.80°$, indicating improved downstream performance with stronger

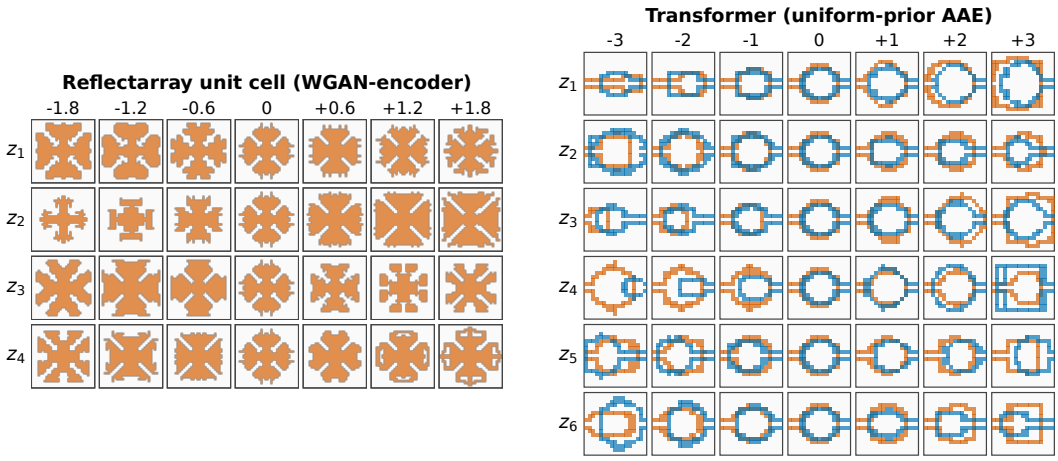
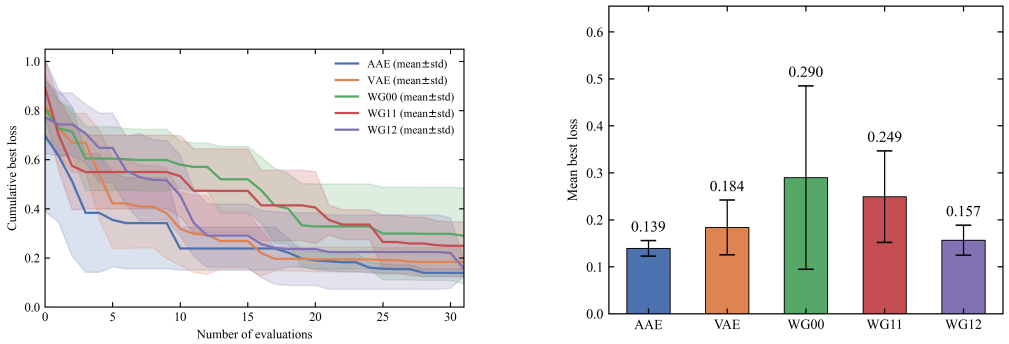


Figure 3: Layouts decoded along latent traversals after training. Each row varies one coordinate z_i (column values) with all other coordinates at zero. Left: reflectarray cells from WGAN-encoder ($z \in \mathbb{R}^4$, traversed over the search bound $[-1.8, 1.8]$), after symmetrization, binarization, and D_4 expansion. Right: two-layer transformer layouts from the uniform-prior AAE ($z \in \mathbb{R}^6$, traversed over the prior support $[-3, 3]$), after projection. All panels are drawn at the same scale; colors follow Figure 2.



(a) Search trajectories

(b) Final best loss

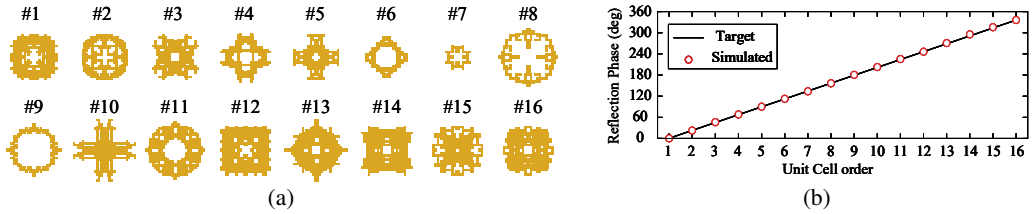
Figure 4: Representation comparison on the reflectarray BO task over five seeds. Curves track the best loss observed during search; shaded bands and error bars denote standard deviations across seeds. Lower loss is preferred. Table 3 reports the final losses and phase errors.

cycle regularization in this setting. Across families, the AAE achieves the lowest mean loss, 0.139, and phase error, $7.57°$, while the VAE reaches 0.184 and $18.89°$. Thus, each representation family supports effective BO, with model choice and regularization affecting convergence and final performance.

4.4 Reflection-phase library synthesis

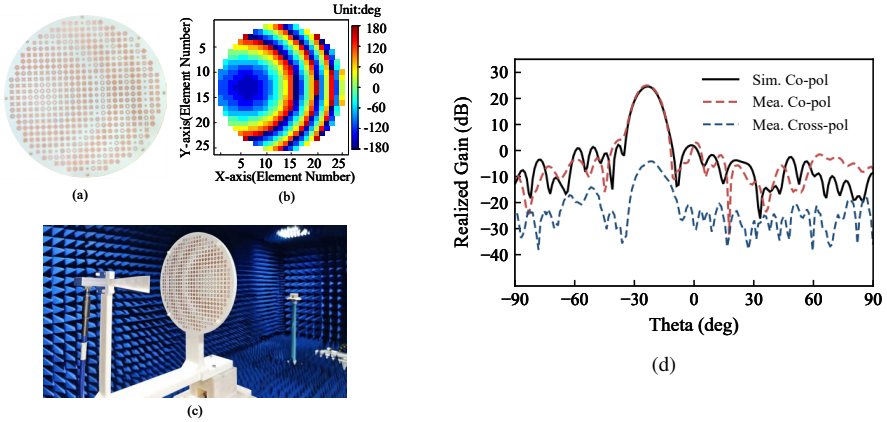
We next examine the complete synthesis task using the encoder-regularized Wasserstein model. Figure 5 shows the resulting 16 layouts and their simulated reflection phases. The library includes ring-like, cross-like, and patch-like patterns within the geometric prior. Their responses follow the requested phase sequence, providing the distinct local phase shifts needed to assemble a reflectarray.

The complete library requires 63 full-wave unit-cell evaluations, including initialization, with observations reused across targets. Model training takes 1.65 h and unit-cell simulations take 3.15 h, totaling approximately 4.8 h for these two stages. The 30,000 training layouts require no EM response



(a) (b)

Figure 5: Optimized reflection library at 10 GHz. (a) Sixteen decoded metal patterns, ordered row by row. (b) Simulated reflection phases and their targets, indexed by the same unit-cell numbers. The target sequence samples the full phase cycle in 22.5° steps. Each layout is obtained through latent search and evaluated by the full-wave solver.



(a) (b) (c) (d)

Figure 6: Reflectarray validation using the optimized unit-cell library. (a) Array configuration. (b) Required phase compensation across the aperture. (c) Fabricated prototype. (d) Simulated and measured radiation patterns at 10 GHz. Co-polarized gain describes radiation in the intended polarization; the measured cross-polarized trace describes the orthogonal component.

labels, concentrating physical evaluation on target-driven synthesis. The 63 evaluations constitute the cumulative budget for all 16 phase targets.

4.5 Fabrication and Validation

To form the array, each location on the aperture is assigned a required reflection phase that compensates for the incident wave's travel distance and directs the reflected beam. The corresponding cell is selected from the optimized library. Figure 6 combines the aperture configuration, phase assignment, fabricated prototype, and measured radiation pattern.

The circular aperture is 250 mm in diameter, and the beam is steered 25° from broadside, that is, from the surface normal. Both simulated and measured main beams point in this direction (Figure 6(d)). The measured peak realized gain, the radiation strength relative to an ideal uniform radiator, is 25.57 dBi at 10 GHz. This measurement validates the use of the optimized unit-cell library in the assembled reflectarray. Array-level validation is separate from the 63 unit-cell evaluations.

4.6 Transformer Synthesis

The uniform-prior AAE is used for transformer optimization with five metric-specific GPs. EM observations accumulate across the 16 targets, and each target uses at most 16 additional EM evaluations. BO improves the objective over each target's initial incumbent in every case. Figure 7 compares requested and achieved quantities for all 16 cases. Many are closely matched. For example, a target of $(80, 50, 0.20)$ is reached at $(80.1, 48.5, 0.198)$, where inductances are in pH.

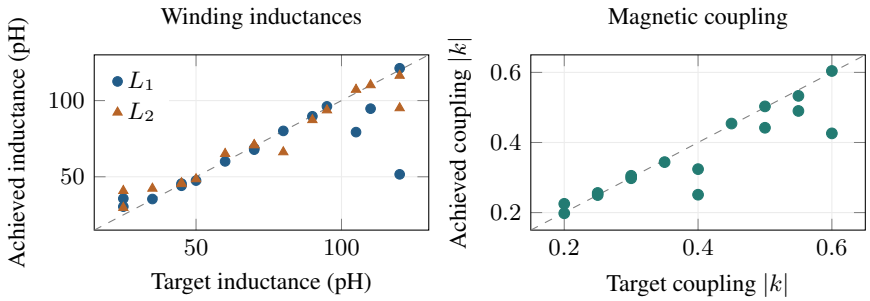


Figure 7: Transformer target matching across all 16 cases. Each point is one target–achieved pair from the EM results; the dashed line denotes an exact match. Both winding inductances are shown on the left, and coupling magnitude on the right.

Larger target deviations occur for very low inductances, large inductance asymmetry, and some combinations of asymmetry and strong coupling. For example, the target $(120, 25, 0.40)$ yields $(51.6, 29.7, 0.324)$. The results support effective synthesis across multiple specifications while motivating broader geometric priors and further search for the more demanding targets.

5 Discussion

The results support learning latent design spaces as an effective approach to pixelated EM optimization. Engineering knowledge defines geometric priors, and representation learning converts these priors into continuous coordinates for search. Four or six latent variables control coordinated changes across an entire layout, reducing the number of optimization variables while retaining flexible pixelated geometries. Geometry-only training requires no EM response labels, concentrating full-wave evaluations on target-driven optimization.

VAE, AAE, and WGAN-encoder all support downstream BO despite their different learning objectives. Model choice and regularization affect convergence and final performance, but successful search is achieved across all three families. The common result is that learned geometric structure provides a useful parameterization for optimization. Reconstruction and generation measures help characterize these representations, while downstream design performance determines their practical value.

The two applications illustrate how this approach accommodates different engineering priors and physical targets. A complete reflection-phase library is obtained with 63 full-wave unit-cell evaluations and validated in a fabricated reflectarray; transformer optimization improves the objective across all 16 targets. These results support the practical value of separating geometric representation learning from physical evaluation. Future work can extend the priors to broader target ranges and quantify simulation savings against direct pixel search under matched budgets.

6 Conclusion

We presented a pipeline for pixelated EM design that combines geometric priors, unsupervised representation learning, and latent-space optimization. Experiments on reflectarrays and on-chip transformers show that compact learned spaces support successful design with limited full-wave evaluations. Multiple representation models provide effective search spaces, supporting the broader approach beyond a single architecture. Learning from inexpensive geometric data therefore offers a practical basis for simulation-efficient optimization of pixelated EM structures.

AI use statement

We used generative AI tools to assist with polishing the manuscript and to modify the layout and presentation of scientific figures. The authors take responsibility for the final content of this work.

REFERENCES

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.20708.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 214–223, 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1704.00028.

Yingqing Guo, Emir Ali Karahan, Zihao Li, Zijian Shao, Zaixi Zhang, Mengdi Wang, and Kaushik Sengupta. Dall-EM: Generative AI with diffusion models for new design space discovery and target-to-electromagnetic structure synthesis. In *2025 IEEE/MTT-S International Microwave Symposium*, pp. 926–929, 2025. doi: 10.1109/IMS40360.2025.11103838. URL https://doi.org/10.1109/IMS40360.2025.11103838.

Emir Ali Karahan, Zheng Liu, Aggraj Gupta, Zijian Shao, Jonathan Zhou, Uday Khankhoje, and Kaushik Sengupta. Deep-learning enabled generalized inverse design of multi-port radio-frequency and sub-terahertz passives and integrated circuits. *Nature Communications*, 15:10734, 2024. doi: 10.1038/s41467-024-54178-1. URL https://www.nature.com/articles/s41467-024-54178-1.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. arXiv:1312.6114, 2013. URL https://arxiv.org/abs/1312.6114.

Zhaxylyk A. Kudyshev, Alexander V. Kildishev, Vladimir M. Shalaev, and Alexandra Boltasseva. Machine learning-assisted global optimization of photonic devices. *Nanophotonics*, 10(1):371–383, 2021. doi: 10.1515/nanoph-2020-0376. URL https://arxiv.org/abs/2007.02205.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. arXiv:1511.05644, 2015. URL https://arxiv.org/abs/1511.05644.

Natalie Maus, Haydn T. Jones, Juston S. Moore, Matt J. Kusner, John Bradshaw, and Jacob R. Gardner. Local latent space Bayesian optimization over structured inputs. arXiv:2201.11872, 2022. URL https://arxiv.org/abs/2201.11872.

Zhichao Tian, Yang Yang, Sui Zhou, Tian Zhou, Ke Deng, Chunlin Ji, Yejun He, and Jun S. Liu. High-dimensional Bayesian optimization for metamaterial design. *Materials Genome Engineering Advances*, 2(4):e79, 2024. doi: 10.1002/mgea.79. URL https://onlinelibrary.wiley.com/doi/10.1002/mgea.79.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In *Advances in Neural Information Processing Systems*, 2020. URL https://papers.nips.cc/paper/2020/hash/81e3225c6ad49623167a4309eb4b2e75-Abstract.html.

Dezhao Zhu, Jiang Guo, Gang Yu, C. Y. Zhao, Hong Wang, and Shenghong Ju. Designing thermal radiation metamaterials via hybrid adversarial autoencoder and Bayesian optimization. *Optics Letters*, 47:3395, 2022. doi: 10.1364/OL.453442. URL https://arxiv.org/abs/2205.01063.

A GEOMETRY-ONLY PRIOR CONSTRUCTION

The reflection-phase library study uses a rule-generated collection of 30,000 layouts with square symmetry, an empty boundary ring, and exclusion of corner-only metallic contacts. The data carry no electromagnetic response labels. Symmetry, fixed regions, and procedural sampling define a structured subset of the full binary pixel space.

The constrained-growth sampler used to illustrate the prior in Figure 2 initializes prescribed metal pixels, protects prescribed empty pixels, and proposes horizontal or vertical three-pixel motifs at neighboring locations. Symmetry expansion produces complete layouts. Occupancy checks, hash-based deduplication, and a Hamming-distance filter control which intermediate layouts are retained.

The models in Table 1 share a training dataset of 20,000 distinct binary 30×30 layouts with square symmetry. Each layout is padded with one empty pixel on each side, and the upper-left 16×16 quadrant forms the model input. This dataset is used for representation evaluation; the phase-library study reports a 30,000-layout training collection.

The transformer dataset contains 50,000 conventional-contour and 50,000 node-controlled layouts, with local motif perturbations. Each accepted layer contains the fixed ports, connects them under an eight-neighbor rule, contains no floating metal, and obeys horizontal reflection symmetry. Finite-area bridges convert the permitted diagonal contacts before simulation. Dataset generation uses no response labels.

B REPRESENTATION OBJECTIVES AND TRAINING CONFIGURATION

The VAE learns a posterior $q_\phi(z \mid x)$ and decoder likelihood $p_\theta(x \mid z)$ through the objective

$$\mathcal{L}_{\mathrm{VAE}} = \mathbb{E}_{x \sim p_{\mathcal{R}}} \left[\mathbb{E}_{z \sim q_\phi(z|x)} [-\log p_\theta(x \mid z)] + \beta \, \mathrm{KL}(q_\phi(z \mid x) \| p(z)) \right]. \tag{5}$$

The likelihood encourages reconstruction, while the KL term regularizes each encoded distribution toward the sampling prior, with strength controlled by β.

For the AAE, the encoder and decoder minimize

$$\mathcal{L}_{E,G}^{\mathrm{AAE}} = \mathbb{E}_x \ell_{\mathrm{rec}}(G_\theta(E_\phi(x)), x) - \lambda_{\mathrm{adv}} \mathbb{E}_x \log D_\psi(E_\phi(x)). \tag{6}$$

For the AAE, the latent discriminator minimizes

$$\mathcal{L}_D = -\mathbb{E}_{z \sim p(z)} \log D_\psi(z) - \mathbb{E}_{x \sim p_{\mathcal{R}}} \log(1 - D_\psi(E_\phi(x))). \tag{7}$$

Together with Equation 6, this regularizes the distribution of encoded layouts toward the sampling prior while retaining a reconstruction objective. The reflectarray VAE and AAE evaluated in Table 1 use four latent coordinates and a residual encoder/decoder architecture. The VAE has separate mean and log-variance heads; both decoders output logits. Both models train for 200 epochs with batch size 512 and Adam learning rate 10^{-4}. Their reconstruction objective is unweighted BCE. The VAE's KL weight β reaches 0.01 after five pretraining epochs and ten warm-up epochs. The AAE uses a standard Gaussian prior, an adversarial weight of 0.05 with the same warm-up schedule, and a discriminator learning rate of 2×10^{-4}.

For the transformer AAE, reconstruction combines binary cross-entropy and Dice loss, and reflected decoder logits are averaged to enforce winding symmetry. The encoder has five convolutional blocks with channel widths 32, 64, 64, 128, and 128, followed by fully connected features of width 256 and six latent outputs. The latent discriminator has two hidden layers of width 64. The main transformer prior is $\mathcal{U}([-3, 3]^6)$. Training uses AdamW, with autoencoder learning rate 10^{-3} and discriminator learning rate 3×10^{-4}. Five reconstruction-only epochs precede a ten-epoch adversarial warm-up.

The WGAN-encoder used for representation evaluation has four latent coordinates and residual up-sampling blocks that produce a 16×16 map. The output uses $\tanh(h/\tau)$, with temperature annealed from 1 to 0.1 over the first 95% of training. Training uses batch size 512 and 200 epochs, with Adam learning rates 10^{-4}, 2×10^{-4}, and 5×10^{-5} for generator, critic, and encoder, respectively, and $(\beta_1, \beta_2) = (0, 0.9)$. Critic updates per generator step change from four to two after epoch 50.

The critic objective is

$$\mathcal{L}_C = \mathbb{E}_z C_\omega(G_\theta(z)) - \mathbb{E}_x C_\omega(x) + 10 \mathbb{E}_{\tilde{x}} (\|\nabla_{\tilde{x}} C_\omega(\tilde{x})\|_2 - 1)^2, \tag{8}$$

where $\tilde{x}$ interpolates real and generated layouts. MMD is estimated with multiple radial-basis-function kernel scales. The phase-library study uses unit MMD and cycle weights, while the model comparison varies them as described in the main text.

B.1 Reflectarray representation evaluation

We sample $N = 1{,}000$ layouts uniformly without replacement from the 20,000-layout training dataset, using random seed 2026 and the same subset for all models. Each 30×30 layout is padded to 32×32, and its upper-left 16×16 quadrant is encoded. The VAE decodes its posterior mean; the AAE and WGAN-encoder use deterministic codes. VAE and AAE inputs lie in $[0, 1]$, whereas WGAN-encoder inputs are scaled to $[-1, 1]$. The WGAN-encoder generator uses temperature $\tau = 1$, and its output is mapped to $[0, 1]$. All models are evaluated with fixed parameters and without geometric projection or repair.

BCE is averaged over all pixels for the VAE and AAE. At probability threshold 0.5, let TP_i, FP_i, and FN_i denote the metal-pixel counts for layout i. The reported overlap scores are

$$\text{Mean IoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}, \qquad \text{Mean F1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2\mathrm{TP}_i}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}. \tag{9}$$

For WGAN-encoder, these scores evaluate $G(E(x))$, although paired reconstruction is not part of its training objective. The reflectarray scores measure training-set reconstruction; the transformer results use held-out layouts and average pooled per-layer overlap scores.

WGAN-encoder is additionally evaluated using its latent-cycle and distribution-matching objectives. We draw $N = 1{,}000$ codes $z_i \sim \mathcal{N}(0, I_4)$ with seed 2026 and compute

$$L_{\mathrm{cyc}} = \frac{1}{Nd}\sum_{i=1}^{N}\|E(G(z_i)) - z_i\|_2^2, \qquad L_{\mathrm{MMD}} = \mathrm{MMD}_k^2\big(\{E(G(z_i))\}_{i=1}^N, \{E(x_i)\}_{i=1}^N\big). \tag{10}$$

Here $d = 4$ and x_i are the same sampled training layouts. MMD^2 is estimated jointly over all 1,000 real and generated codes using the adaptive multiscale RBF kernel and biased estimator used during training, including diagonal terms. These two losses assess latent-code recovery and encoded-distribution alignment, respectively.

C Reflection model-comparison summaries

Table 3 summarizes the five-seed comparison in Figure 4. This experiment uses BO with expected improvement and 16 initial points. It evaluates variation in search outcomes across representation models separately from the cumulative phase-library synthesis experiment.

Table 3: Five-seed reflection BO comparison, mean $\pm$ standard deviation. Lower values are preferred. WG labels specify MMD and cycle weights.

Representation	Optimization loss	Phase error (degrees)
WG00	0.290 ± 0.195	42.65 ± 43.97
WG11	0.249 ± 0.098	38.69 ± 26.83
WG12	0.157 ± 0.032	10.80 ± 7.27
VAE	0.184 ± 0.065	18.89 ± 21.43
AAE	$\mathbf{0.139 \pm 0.016}$	$\mathbf{7.57 \pm 3.10}$

D Reflectarray simulation configuration

Phase is circular: 0° and 360° describe the same response. We therefore use the circular error

$$\Delta(\psi, \phi) = |((\psi - \phi + 180) \bmod 360) - 180|, \tag{11}$$

where ψ is the simulated phase and ϕ the target. The library objective is

$$J(x;\phi) = \sum_{i\in\{0,-,+\}} w_i \frac{\Delta(\psi(x;f_i),\phi)}{180}, \tag{12}$$

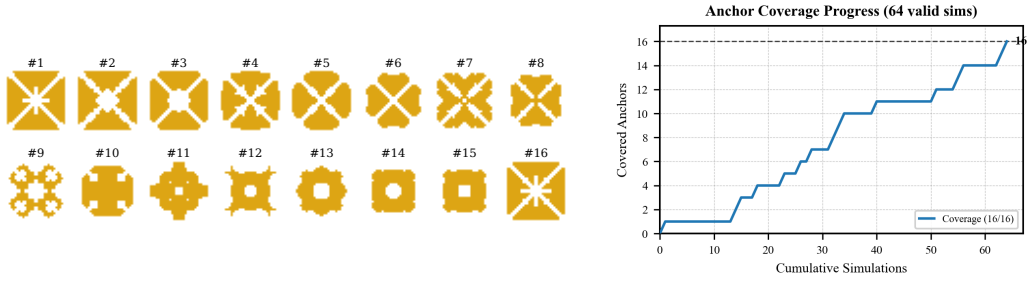
with $(f_0, f_-, f_+) = (10, 9.5, 10.5)$ GHz and weights $(0.6, 0.2, 0.2)$. Sixteen targets, spaced by $22.5°$, cover the phase cycle. The stopping criterion is center-frequency error below $5°$.

The unit cell uses Rogers 4003C with relative permittivity 3.55, loss tangent 0.0027, thickness 1.524 mm, and periodicity 10 mm. A continuous copper ground plane backs the substrate. Full-wave evaluation uses HFSS. The reported workstation has an Intel Core i5-13600K processor at 3.50 GHz, 64 GB RAM, and a graphics card with 8 GB VRAM.

Library construction uses a Matérn-5/2 GP and LogEI with the objective in Equation 12. Sixteen initialization candidates populate an empty database, which is subsequently shared across phase targets. The reported 63 full-wave unit-cell evaluations refer to this cumulative library run. The array-level radiation results in Figure 6 are separate from that budget.

D.1 AAE-BASED REFLECTION-PHASE LIBRARY SYNTHESIS

An additional BO run in the four-dimensional AAE space constructs a reflection-phase library for the 16 targets spaced by $22.5°$. In this run, a target is covered when a simulated unit cell has circular phase error below $6°$ and reflection amplitude $|S_{11}| > 0.8$ at 10 GHz. The shared simulation database contains 64 distinct, valid unit cells and covers all 16 targets. Figure 8 shows the resulting library and the cumulative target coverage. For each target, the displayed unit has the smallest phase error among the simulated candidates satisfying the amplitude criterion. This result provides an additional example of complete phase-library synthesis using an autoencoder-based design space.



(a) (b)

Figure 8: Reflectarray library synthesis using AAE-based BO. (a) Sixteen optimized unit cells in a 2×8 arrangement. Units #1–#16 correspond to target phases from $0°$ to $337.5°$ in $22.5°$ steps, ordered row by row. (b) Number of covered target phases (anchors) versus cumulative simulations. All 16 targets are covered after 64 simulations under the $6°$ phase-error and $|S_{11}| > 0.8$ criteria at 10 GHz.

E TRANSFORMER OBJECTIVE AND COMPLETE RESULTS

For metrics $y = (L_1, L_2, |k|, Q_1, Q_2)$ and targets $t = (L_1^\star, L_2^\star, |k|^\star)$, the reported objective is

$$J(y;t) = \sum_{m\in\{L_1,L_2,|k|\}} w_m \left(\frac{y_m - t_m}{t_m}\right)^2 + \frac{w_{Q_1}}{1+\max(Q_1,0)} + \frac{w_{Q_2}}{1+\max(Q_2,0)}. \tag{13}$$

The quality-factor terms favor lower dissipation, while the squared relative errors target the requested inductances and coupling. The weights w_m, w_{Q_1}, and w_{Q_2} control these preferences. Nine of the 16 cases achieve a final objective below 0.05; Table 4 reports the physical metrics for each case.

Each metric is modeled by a separate GP, and Monte Carlo posterior samples define the utility $-J$ for qLogEI. One candidate is evaluated at a time. Invalid decodes receive bounded latent perturbations before retrying. EM observations are reused across successive targets, with at most 16 additional evaluations per target.

Table 4: All transformer target and achieved metrics. Inductances are in pH. These values also generate Figure 7.

Case	Target L_1/L_2	Target $\lvert k\rvert$	Achieved L_1/L_2	Achieved $\lvert k\rvert$
1	25 / 25	.20	35.7 / 40.8	.225
2	120 / 120	.60	121.2 / 116.4	.604
3	25 / 120	.40	30.4 / 95.1	.251
4	120 / 25	.40	51.6 / 29.7	.324
5	50 / 80	.60	47.5 / 66.2	.426
6	80 / 50	.20	80.1 / 48.5	.198
7	45 / 105	.30	44.1 / 107.2	.305
8	105 / 45	.50	79.3 / 45.9	.442
9	70 / 95	.25	68.7 / 93.7	.250
10	95 / 70	.55	96.2 / 70.8	.533
11	35 / 70	.25	35.4 / 70.9	.256
12	70 / 35	.55	67.8 / 42.3	.490
13	45 / 90	.45	45.5 / 87.3	.454
14	90 / 45	.35	89.7 / 45.3	.344
15	60 / 110	.30	60.1 / 110.2	.298
16	110 / 60	.50	94.7 / 65.1	.503