FENCE: ENFORCING MIXED FUNCTIONAL CONSTRAINTS IN NEURAL FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Enforcing constraints at sampled points does not guarantee that a neural field remains admissible throughout the domain. We introduce Functional Equality–iNequality Constraint Enforcement (FENCE), which assembles linear field functionals, basis envelopes, and cover-dependent variation margins into a finite convex feasible set enforced by a standard differentiable QP layer. For representations with valid enclosure or variation bounds, this jointly enforces equalities and continuous-domain inequalities without a nonlinear inner solve. On heat design, sufficient constraints pass all independent dense and finite-difference checks across held-out tasks and seeds. Constraint exchange preserves full-system objectives while cutting final QP size by over two orders of magnitude on a development cover sweep. On motion retargeting, derivative envelopes meet continuous rate limits on every held-out clip where sampling misses inter-sample violations, with $< 0.1\%$ mean objective overhead relative to analytic exchange. Matched QP implementations confirm that coverage follows from the constraint construction.

1 INTRODUCTION

A neural field that matches observations and satisfies every sampled constraint can still exceed a physical limit between samples. This matters when a heat source must remain nonnegative throughout a domain, a protected region must stay below a temperature cap, or a trajectory must preserve keyframes while respecting rate limits. The constraint description, not the penalty weight, determines which coefficients are admissible and what part of the field is covered (Krishnapriyan et al., 2021; Rathore et al., 2024).

Constrained Neural Fields (CNF) map linear field functionals into coefficient equalities $A\beta = b$ (Zhong et al., 2023). Differentiable QP layers already handle mixed algebraic constraints (Amos & Kolter, 2017; Agrawal et al., 2019). Combining them raises a coverage question: the description must capture both observation equalities and bounds that remain valid over the intended domain. Sufficient bounds can exclude useful designs, so we evaluate coverage together with conservatism and cost.

We introduce **Functional Equality–iNequality Constraint Enforcement (FENCE)**: a mixed field-constraint description for $f_\beta(x) = \Psi(x)^\top \beta$, followed by coefficient-space projection. This representation connects predicted coefficients to observation equalities and physical bounds in one differentiable layer. It lets an encoder fit observations while the projection enforces a feasible coefficient set $\mathcal{C}$. Point evaluations, derivatives, and integrals supply linear rows. Where the representation permits it, nonnegative basis envelopes and a cover radius times a field-variation bound supply *sufficient* rows for continuous-domain inequalities. These rows compose with the equalities in one convex feasible set; a standard differentiable QP enforces the resulting contract.

Our contributions are:

- A formulation that separates equality–inequality composition from continuous coverage, with explicit assumptions for turning basis envelopes and cover margins into finite affine constraints.
- Concrete spatial and temporal contracts: source and protected-temperature bounds for heat design, and whole-interval derivative bounds for cubic-spline motion, with measured coverage–quality and constraint-count tradeoffs.

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.

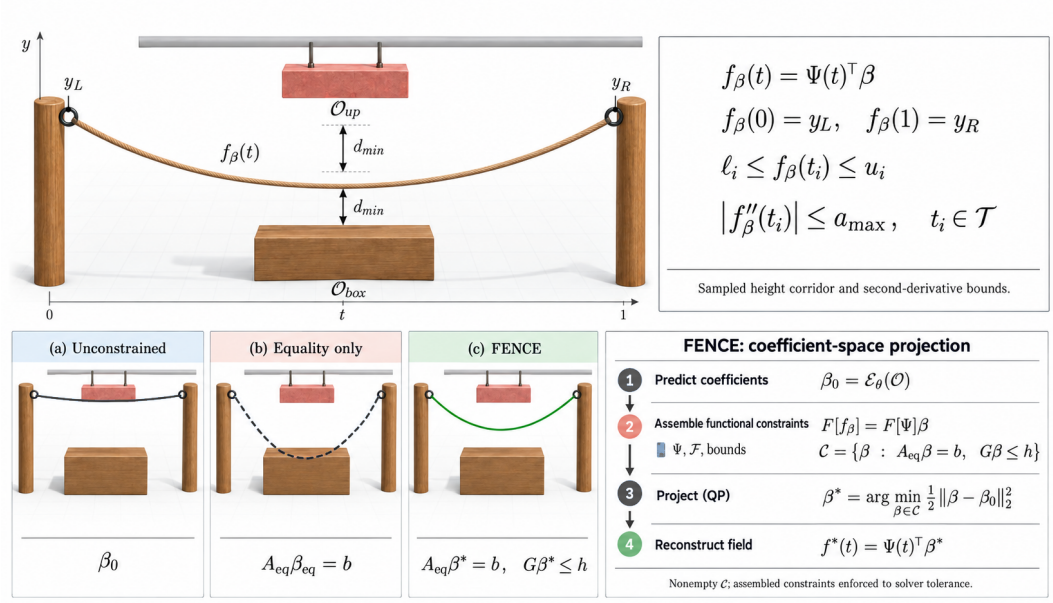


Figure 1: **FENCE pipeline.** Schematic of a 1-D field with fixed endpoints, sampled height-corridor bounds, and second-derivative bounds (not geometric curvature). Left panels illustrate a sampled-bound setting; continuous coverage uses the sufficient constructions in Section 3.3, evaluated in Section 4.2. An encoder predicts β_0; linear functionals assemble A_{eq}, b, G, h; a QP returns $\beta^\star = \Pi_{\mathcal{C}}(\beta_0)$; the field is $f^\star(t) = \Psi(t)^\top \beta^\star$. Assembled linear constraints hold to solver tolerance; sufficient constructions cover the continuum under their stated assumptions.

- Empirical comparisons showing that sufficient constructions pass independent dense checks where sampling fails, that finer covers reduce conservatism at higher row counts, and that constraint exchange shrinks the final QP without changing the finite feasible set. Matched QP and direct-task baselines separate effects of the constraint construction from the choice of solver or predictor.

2 Related Work

Constraint enforcement. Physics-informed losses encode residuals as penalties (Raissi et al., 2019); adaptive weighting and augmented Lagrangians improve optimization (Son et al., 2023; Wang et al., 2021). Finite penalties generally trade residual against task loss (Nocedal & Wright, 2006). Hard boundary constructions (Lu et al., 2021) and null-space parameterizations (Hendriks et al., 2020) enforce selected conditions by design. DC3 combines equality completion with iterative inequality correction (Donti et al., 2021), while HardNet and KKT-based layers enforce constraints on network outputs (Min & Azizan, 2024; Chen et al., 2024). Separating the constraint representation from its numerical enforcement is a recurring theme (Négiar et al., 2023; Wang et al., 2023; Constante et al., 2025).

Differentiable optimization. OptNet (Amos & Kolter, 2017), differentiable convex layers (Agrawal et al., 2019), and deep declarative networks (Gould et al., 2022) provide implicit differentiation through constrained problems (Blondel et al., 2022). FENCE uses standard QP and active-face derivatives (Magoon et al., 2025), focusing on assembly from field functionals. A QP layer with identical rows has the same feasible set; our experiments include this control.

Continuous and semi-infinite constraints. Enforcing an inequality over a continuum is a semi-infinite program (Hettich & Kortanek, 1993). Exchange methods add violated constraints iteratively; conservative inner approximations replace each continuum constraint by a finite sufficient condition. The Bernstein convex-hull enclosure used here belongs to the second family. Our contribution is

Table 1: **Constraint interfaces and enforcement.** FENCE combines observation equalities and physical bounds in one coefficient-space feasible set. The field interface expresses the connection between these constraints and the underlying representation; feasibility is evaluated at solver tolerance.

Method	Equality	Inequality	Field interface	Feasibility
Penalty / PINN	soft	soft	task residuals	not enforced
DC3	exact	approximate	algebraic outputs	equality
HardNet	optional	exact	algebraic outputs	supported set
OptNet / Diff-QP	exact	exact	algebraic QP rows	assembled set
LCNN	homogeneous	—	linear output operators	equality
CNF	exact	—	equality functionals	assembled set
FENCE	exact	exact	mixed functionals	assembled set

composing that enclosure with equality functionals in one coefficient-space convex set, and measuring how conservatism and row count trade off against cover density and subdivision. Continuous coverage comes from the enclosure; exchange is only a constraint-generation device.

Constrained neural fields. Coordinate networks and kernel expansions provide continuous representations (Williams et al., 2021; Sitzmann et al., 2020; Tancik et al., 2020; Park et al., 2019; Williams et al., 2022; Gropp et al., 2020; Fathony et al., 2021). CNF (Zhong et al., 2023) compiles linear operators on a field into coefficient equalities. We retain that view and add inequality rows for bounds, budgets, and exclusion regions. Table 1 summarizes the interfaces.

Application context. Density-based topology optimization (Bendsøe & Sigmund, 2003) and neural topology representations (Zehnder et al., 2021) support specialized design procedures. Retargeting and motion tracking address embodiment constraints (Araujo et al., 2025; Aberman et al., 2020). Our comparisons evaluate the field-constraint description within these pipelines.

3 METHODS

We introduce FENCE, combining mixed affine rows with explicit coverage assumptions. Applications differ in the basis, predicted coefficients, and assembled functionals (Figure 1):

$$\begin{aligned} f_{\beta_0}(x) &\xrightarrow{\mathcal{F}[\Psi]} \mathcal{C} = \{\beta : A\beta = b,\ G\beta \le h\}, \\ \beta^\star = \Pi_{\mathcal{C}}(\beta_0) &\xrightarrow{f_{\beta^\star}(x)=\Psi(x)^\top\beta^\star} f^\star. \end{aligned} \tag{1}$$

The middle stages assemble explicit linear field operators into coefficient-space affine rows; the projection itself is a standard convex QP (as in OptNet-style layers). Our focus is the constraint description and its coverage; differentiation follows existing optimization layers.

3.1 NEURAL FIELDS AS COEFFICIENT-SPACE OBJECTS

Let $\Omega \subset \mathbb{R}^d$ be a spatial, temporal, or spatio-temporal domain. We represent a scalar or vector-valued field by a basis expansion

$$\begin{aligned} f_\beta(x) &= \Psi(x)^\top \beta, \\ \Psi(x) &= [\Psi_1(x), \dots, \Psi_M(x)]^\top. \end{aligned} \tag{2}$$

The experiments use fixed evaluation bases. When a basis is learned, the reported backward pass holds (A, G, b, h) fixed; it omits assembly derivatives. A neural encoder E_θ predicts unconstrained coefficients

$$\beta_0 = E_\theta(\mathcal{O}), \tag{3}$$

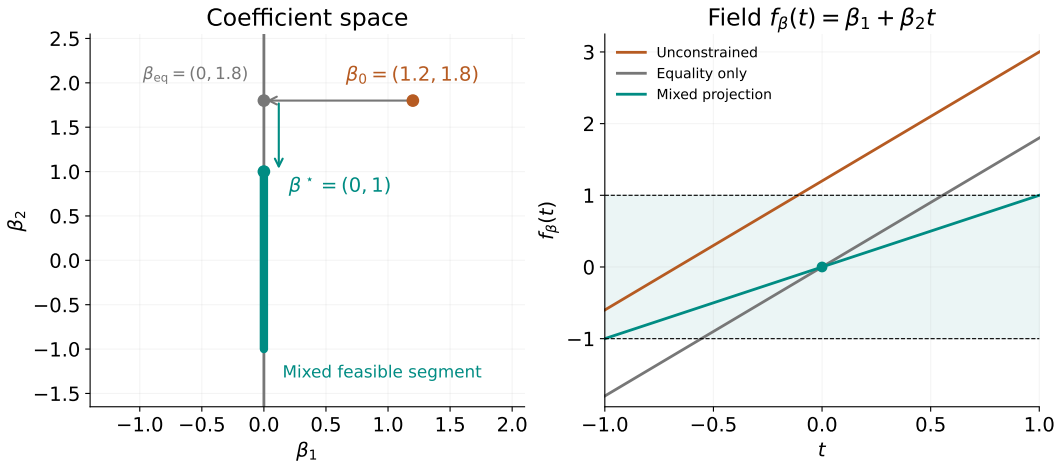


Figure 2: **Equality manifold versus mixed feasible set.** Toy field $f_\beta(t) = \beta_1 + \beta_2 t$ with equality $f_\beta(0) = 0$ and bound $|f_\beta(t)| \le 1$ on $[-1, 1]$. Equality-only projection yields $(0, 1.8)$ (outside the bound); mixed projection yields $(0, 1)$ and meets both requirements. Endpoint bounds suffice for this affine field (schematic, not an experiment).

from observations $\mathcal{O}$ such as partial trajectories, keypoints, scans, or physical measurements. The role of FENCE is to map β_0 to feasible coefficients $\beta^\star$ before the field is queried or optimized.

We use the term *functional constraint* broadly. A functional may be a point evaluation, boundary trace, differential operator, integral, or linearized physical limit. For a linear functional $\mathcal{F}$,

$$\mathcal{F}[f_\beta] = \mathcal{F}[\Psi^\top \beta] = \mathcal{F}[\Psi]\,\beta. \tag{4}$$

The interface composes basis evaluations, derivative operators, and integral rows with equality and bound declarations. Applications supply linearity and coverage arguments. Appendix M specifies the shared assembly API used by heat training and the continuous-motion extension; earlier experiments use task-specific assembly.

3.2 Mixed Equality–Inequality Functional Constraints

We consider feasible sets of the form

$$\mathcal{C} = \left\{ \beta \in \mathbb{R}^M \;\middle|\; \begin{array}{l} A_{\mathrm{eq}}\beta = b, \\ A_{\mathrm{lb}}\beta \ge \ell, \\ A_{\mathrm{ub}}\beta \le u \end{array} \right\}. \tag{5}$$

Linear interpolation, boundary values, and linear conservation or PDE residuals form equality rows. Linear bounds, integral budgets, and sampled void requirements form inequality rows. Nonlinear dynamics, kinematic closure, stress, and clearance constraints require additional linearization or a separate solver; they are not linear functionals in general.

The feasible field is obtained by Euclidean projection:

$$\boxed{\begin{array}{l} \beta^\star = \Pi_{\mathcal{C}}(\beta_0) = \arg\min\limits_{\beta} \; \dfrac{1}{2}\|\beta - \beta_0\|_2^2 \\ \quad \text{s.t.} \quad A_{\mathrm{eq}}\beta = b, \\ \qquad\quad A_{\mathrm{lb}}\beta \ge \ell, \qquad A_{\mathrm{ub}}\beta \le u. \end{array}} \tag{6}$$

The final prediction is $f^\star(x) = \Psi(x)^\top \beta^\star$. When A_{lb} and A_{ub} are absent, Eq. (6) reduces to the equality projection used by CNF. With inequalities present, it projects onto a feasible region rather than an affine equality manifold.

3.3 Finite Constraints with Continuous Coverage

Pointwise rows and continuous coverage define different contracts. We construct sufficient affine rows with two standard bounding principles rather than treating dense samples as a guarantee.

Basis envelopes. Suppose

$$f(x) = s(x) \sum_j \gamma_j B_j(x), \qquad 0 \le s \le 1, \quad B_j \ge 0, \quad \sum_j B_j = 1.$$

Then $0 \le \gamma_j \le u$ suffices for $0 \le f(x) \le u$ everywhere (Farouki, 2012). If $\gamma = L\beta$, these are linear coefficient inequalities. The converse need not hold: the enclosure can exclude admissible fields. For heat, the identity

$$\sin(m\pi x) = \sin(\pi x) U_{m-1}(\cos \pi x)$$

converts the fixed sine expansion into a tensor-product Bernstein form, yielding a concrete invertible map $z = C\gamma$. For cubic topology fields, nonnegative B-spline weights give the analogous bound (de Boor, 1978); the analytic void cut is shared by all methods and is separate from projection. For motion, q' on each cubic-spline span is a quadratic Bernstein polynomial with coefficients linear in β. Bounding these three coefficients enforces the speed limit over the span; subdivision reduces conservatism. Linear acceleration needs only endpoint bounds (Appendix O).

Cover-dependent variation bounds. Let a convex region D have a finite δ-cover $\{x_i\}$. Suppose

$$\gamma \ge 0, \qquad T(x) = a(x)^\top \gamma, \qquad \sup_{x \in D} \|\nabla T(x)\|_2 \le w^\top \gamma.$$

Then the affine inequalities

$$\begin{aligned} [a(x_i) + \delta w]^\top \gamma \le T_{\text{safe}} \quad &\text{for every } i \\ \implies \quad T(x) \le T_{\text{safe}} \quad &\forall x \in D. \end{aligned} \tag{7}$$

Indeed, every x lies within δ of a cover node; integrating the gradient along their segment gives $T(x) \le T(x_i) + \delta w^\top \gamma$. A row residual ϵ relaxes the conclusion by the same ϵ. The assumption must hold on the connecting segments.

For the Poisson sine field,

$$\begin{aligned} a(x)^\top &= \psi(x)^\top \operatorname{diag}(\lambda)^{-1} C, \\ w^\top &= [\pi \sqrt{m^2 + n^2}/\lambda_{mn}]_{mn}^\top |C|. \end{aligned}$$

Nonnegativity of γ turns the triangle-inequality gradient bound into a linear form. A polar cover of a disk of radius r gives $\delta \le r(1/(2n_r) + \pi/n_a)$. Thus source bounds, disk-temperature bounds, and the analytic power budget remain one finite convex contract. Smaller δ reduces the margin but requires more rows. Appendix I gives the complete construction; this bound applies to the continuum field, with FD validation separate.

Constraint exchange. To avoid a large QP, solve with a subset of inequality rows, scan the full finite system, and add the most violated rows. If the subset optimum satisfies every row, it is also the full-system optimum: the full feasible set is a subset containing that optimum. We check all rows before accepting a solution and record budget failures. Continuous coverage comes from Eq. (7), not from this standard finite constraint-generation rule (Hettich & Kortanek, 1993).

3.4 Projection and Differentiation

Writing the inequalities as $G\beta \le h$, the standard projection QP and KKT conditions are (Boyd & Vandenberghe, 2004)

$$\min_\beta \ \tfrac{1}{2}\beta^\top \beta - \beta_0^\top \beta \quad \text{s.t.} \quad A\beta = b, \quad G\beta \le h. \tag{8}$$

$$\begin{aligned} \beta - \beta_0 + A^\top \nu + G^\top \lambda = 0, \\ A\beta = b, \qquad G\beta \le h, \\ \lambda \ge 0, \qquad \lambda \odot (G\beta - h) = 0. \end{aligned} \tag{9}$$

For a locally constant active set, stack $\tilde{A} = [A^\top, G_{\mathcal{A}}^\top]^\top$. The fixed-matrix Jacobian and backward pass for upstream gradient r are

$$P_{\tilde{A}} = I - \tilde{A}^\top (\tilde{A}\tilde{A}^\top)^\dagger \tilde{A}, \tag{10}$$

$$\partial \mathcal{L}/\partial \beta_0 = P_{\tilde{A}} r. \tag{11}$$

Active-set transitions can be nonsmooth (Khanduri et al., 2023). Heat derivatives are checked by finite differences and an independent QP implementation.

Projection onto a nonempty closed convex $\mathcal{C}$ is unique; numerical feasibility is set by solver tolerance (Appendix G). Continuous claims require a valid enclosure or variation bound (Section 3.3). Euclidean projection minimizes coefficient displacement; a field-space L^2 metric instead uses the basis Gram matrix. Nonlinear contact, stress, and kinematic constraints need an outer procedure and validation in the original model (Appendix D).

4 Evaluation

We evaluate constraint composition and continuous coverage on motion correction, heat design, and topology. Independent held-out checks are separated from development diagnostics.

4.1 Motion Retargeting with Discrete and Continuous Rate Constraints

We correct GMR (Araujo et al., 2025) trajectories for Unitree G1 using LAFAN1 (Harvey et al., 2020). Equalities preserve 14 upper-body joint coordinates at five uniformly spaced frames of each *unmodified* five-second trajectory. Inequalities bound first and second differences at 30 Hz by 2.5 rad/s and 20 rad/s^2 (experimental limits, not hardware ratings). Equality-only projection leaves the raw trajectory unchanged. Figure 3 shows four development excerpts: FENCE keeps the anchors and meets both rate limits (mean RMS change 1.78°/joint).

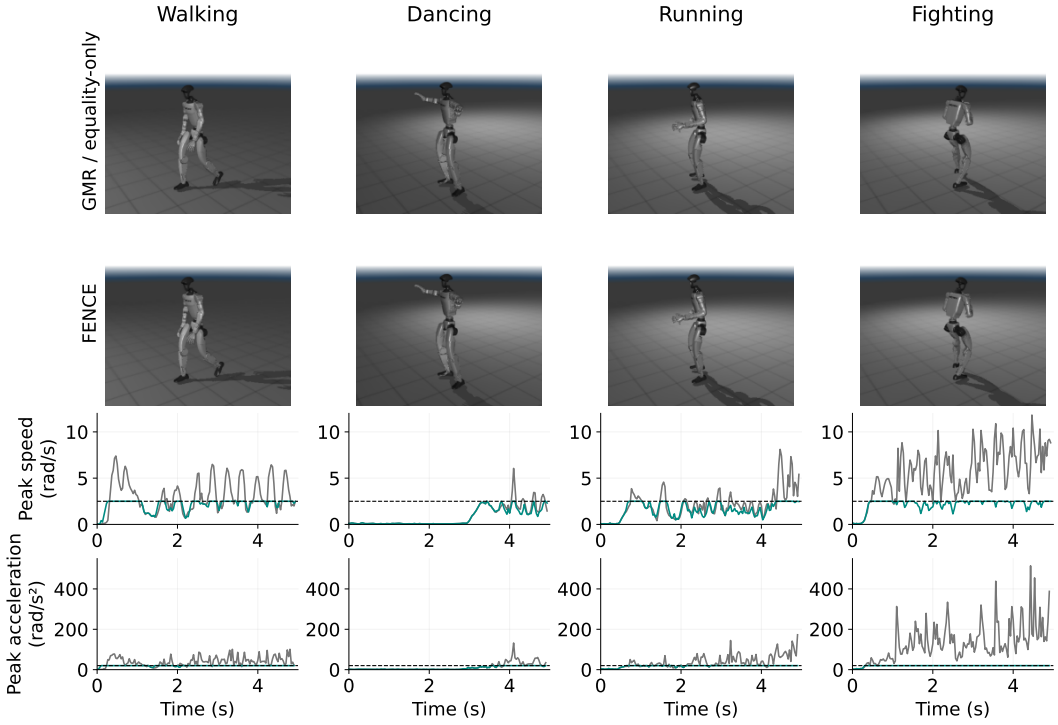


Figure 3: **Discrete rate correction on four LAFAN1 excerpts.** Columns (left to right): Walking, Dancing, Running, Fighting. Top two rows: GMR/equality-only versus FENCE at the same camera and time t=2.5 s. Bottom two rows: peak absolute joint speed (rad/s) and acceleration (rad/s^2) over the five-second window (gray: GMR/equality-only; teal: FENCE); dashed lines mark the 2.5 rad/s and 20 rad/s^2 limits. FENCE preserves raw keyframes while clipping discrete rates.

On ten held-out excerpts from subjects 3–5 (selection frozen on development data), FENCE and an independent direct QP each satisfy the full discrete contract in 10/10 cases (mean RMS 2.24°). A soft quadratic penalty and filtering each pass only 1/10 at tolerance 10^{-5}; their largest speed and acceleration residuals are 1.0×10^{-4} rad/s and 3.6×10^{-4} rad/s^2. Appendix J.1 lists excerpts and

renderings; Appendix J discusses GMR's native IK velocity controls. These offline results concern joint-coordinate rates only.

A separate cubic-spline extension bounds actual time derivatives (Appendix O). After development on four clips, seven new clips give 7/7 passes for Bernstein envelopes and analytic exchange, versus 0/7 for 30/120-Hz sampling (Table 2), at $7.3 \times 10^{-2}\%$ mean objective overhead relative to exchange. Envelope and Direct QP constructions both pass; wall-clock differences depend on the clip set (Table 2 vs. Appendix O).

Table 2: **Frozen continuous-time validation.** Seven new clips with identical physical thresholds and fitting objective. Pass requires speed excess $\leq 10^{-6}$ over every spline interval. Columns: construction, pass count, mean RMS pose change, max speed excess (rad/s), and wall-clock seconds (Appendix O).

Construction	Pass	Mean RMS (°)	Max speed excess	Seconds
Sampled 30 Hz	0/7	2.56	0.167	0.22
Sampled 120 Hz	0/7	2.57	8.2×10^{-3}	0.63
FENCE (one-span envelope)	7/7	2.57	4.3×10^{-10}	0.27
FENCE (two subdivisions)	7/7	2.57	2.9×10^{-10}	0.39
FENCE (four subdivisions)	7/7	2.57	1.7×10^{-10}	1.22
FENCE (analytic exchange)	7/7	2.57	2.4×10^{-8}	0.68
Direct task QP (one-span)	7/7	2.57	3.9×10^{-13}	0.21

4.2 Heat Design: From Sampled Bounds to Continuous Coverage

We design a source q to match a desired temperature while protecting a disk. The physical equality is $-0.03\Delta T = q$ on the unit square with zero boundary temperature; limits are $0 \leq q \leq 2, T \leq T_{\text{safe}}$ on the disk, and $\int q \leq P_{\max}$. A 4×4 sine expansion couples source and temperature through Poisson eigenvalues.

Sampled versus continuous coverage. Assembling source bounds at 8×8 points and temperature bounds at 17 disk points yields an assembled-feasible QP that still covers only 1.4% of independent dense/FD checks (0% under tighter limits); a direct sampled optimum behaves the same (Appendix H, Table 5, Figure 8). Coverage, not the solver, is the failure mode.

Sufficient constraints on independent tasks. Section 3.3's Bernstein source bounds, power row, and temperature cover margins are given to every method. We train a 9–64–64–32 encoder on 128 tasks (200 Adam steps, batch 8), select hyperparameters on 32 validation tasks, then evaluate three seeds on 96 held-out tasks. CNF-style penalties and adapted DC3 test soft enforcement; OptNet matches the projection with a separate backward pass; Direct QP optimizes the task objective under the same rows (Appendix I). Both QP layers pass all 288 dense/FD checks (Table 3). FENCE and

Table 3: **Frozen sufficient-constraint heat test.** Mean ± sample std. over three training seeds × 96 held-out tasks (N=288). Full-row and dense/FD pass rates use tolerance 10^{-6}. J is the task objective (lower better); distinct from Table 5.

Method	$10^3 J$	Full sufficient rows (%)	Dense/FD (%)
CNF-style + penalty	10.356 ± 0.634	0.0	8.7
DC3 (adapted)	12.679 ± 0.141	21.2	98.6
OptNet	$\underline{8.507} \pm \underline{0.276}$	**100.0**	**100.0**
FENCE	$\mathbf{8.163 \pm 0.195}$	**100.0**	**100.0**
Direct task QP	6.774	100.0	100.0

OptNet attain comparable mean task objectives (8.163 versus 8.507) with identical 100% sufficient-row and dense/FD pass rates, while Direct task QP reaches a lower objective (6.774) as a direct optimizer with full task information. Relative to the CNF-style penalty adaptation, FENCE raises dense/FD from 8.7% to 100%. Figure 4 shows the first task and seed.

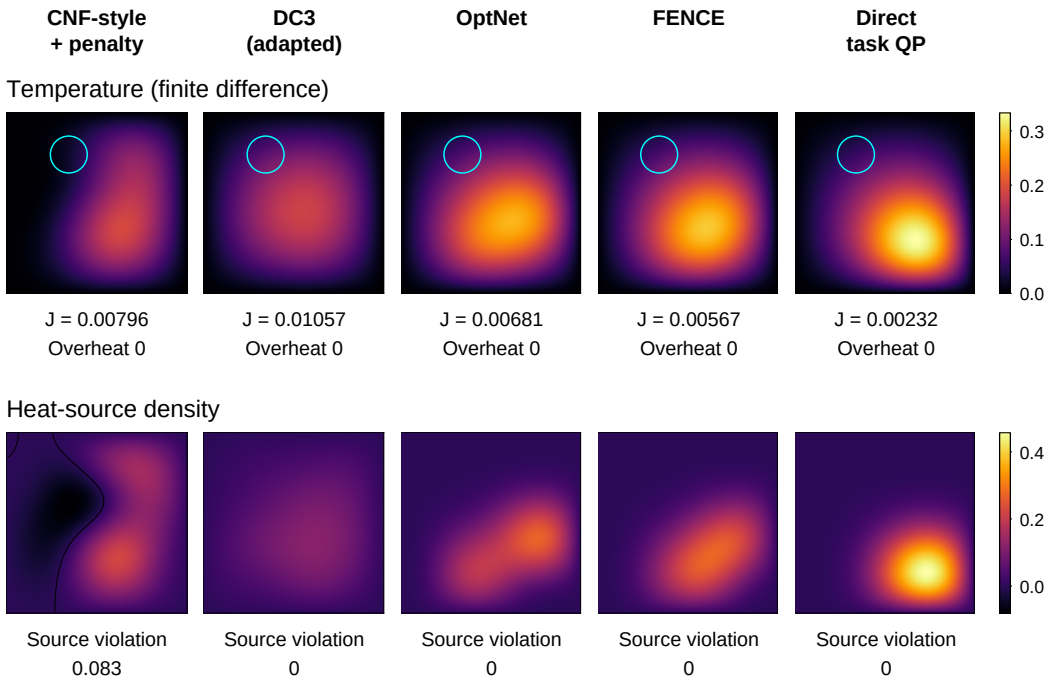


Figure 4: **Sufficient-constraint heat comparison on one held-out task.** Independent test case 0 (generation seed 5300) under training seed 11; every method receives the same sufficient rows. Columns: CNF-style+penalty, adapted DC3, OptNet, FENCE, Direct QP. Top: finite-difference temperature (cyan: protected disk); bottom: heat-source density. Per-panel J, overheat, and source-violation annotations describe this case, not an average advantage. Aggregate objectives and pass rates are reported in Table 3 ($10^3 J$); this figure reports unscaled J.

Coverage, quality, and cost. Holding Bernstein source and power rows fixed on 12 development tasks, finer temperature covers reduce conservatism; the margin reaches 12/12 at every density (Appendix N, Table 11). Exchange cuts the densest QP from 16,673 rows to $\leq$89 ($187\times$) at identical objectives. Adding a source-centroid equality raises mean J from 5.50×10^{-3} to 8.34×10^{-3} while keeping all dense/FD passes. Frozen predict+project takes 7.69 ms median versus 7.57 ms for direct optimization on the same tasks (excluding common assembly): at this scale we observe no inference speedup from the learned encoder.

4.3 Topology Design: Contracts and Coverage

Three development loads use cubic B-splines with shared analytic voids, volume 0.4, 64×24 knot spans, and 192×72 FEM cells. Nonnegative basis weights extend coefficient bounds to the field. Neural controls share filtered OC targets (Zehnder et al., 2021), 40 physical updates, and 200 fitting steps. Figure 5 contrasts equality-only projection with a bound penalty (CNF-style) against mixed projection (FENCE). Across three seeds on these fixed geometries, FENCE passes 9/9 complete contracts versus 1/9 for CNF-style (Table 10). On the recorded geometries, Direct OC reaches lower compliance on tip and shear, while the QP methods are competitive on half-MBB (Appendix L).

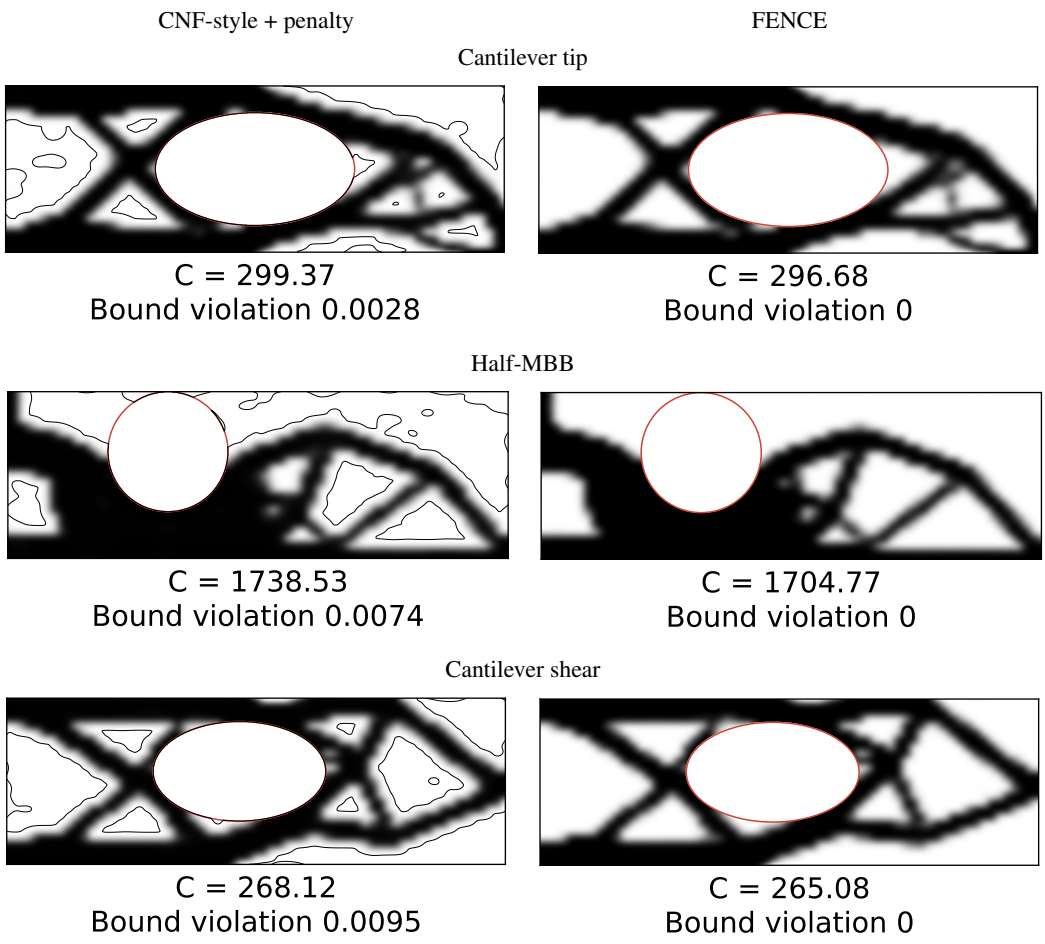


Figure 5: **Hard bound enforcement versus a penalty on three development geometries.** Left: CNF-style + penalty; right: FENCE; rows: cantilever tip, half-MBB, cantilever shear (seed 41). Each panel evaluates the saved field on a 1536×512 grid (shared gray scale); red marks the shared analytic void. Annotations report compliance C (lower better) and max $[0, 1]$ field-bound violation before the common mechanical extension. FENCE meets the bounds in every shown case; the penalized control leaves residual violations.

5 Discussion and Conclusion

FENCE connects learned field coefficients, observation equalities, and domain-wide bounds through a differentiable projection. Its value is a shared representation in which fitting observations and enforcing physical limits can be composed during learning. On heat design, sufficient rows achieve 100% dense/FD passes with objectives comparable to OptNet; direct task optimization attains lower objectives. On cubic-spline motion, derivative envelopes close inter-sample gaps with less than 0.1% mean objective overhead relative to analytic exchange.

Limitations and outlook. The experiments support constraint composition and continuous-domain bounds under explicit basis assumptions. Coarse covers can be conservative, and the current encoder provides no consistent inference speedup. A useful next step is to evaluate whether learned priors help when observations are incomplete, while retaining the same projection for physical bounds. This would require broader tests than the present development sets and extensions for nonlinear contact and torque constraints. Physical robot frames in Appendix A remain qualitative; closed-loop control is not evaluated here.

AI USE STATEMENT

AI tools were used to assist with mathematical formulations and code development, as well as manuscript editing and formatting. The authors are responsible for the accuracy and integrity of the mathematical content, code, experiments, and conclusions.

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-Aware Networks for Deep Motion Retargeting. *ACM Transactions on Graphics (TOG)*, 39(4):62, 2020. URL https://deepmotionediting.github.io/retargeting.

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pp. 136–145, 2017.

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

Martin P. Bendsøe and Ole Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer, 2003.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-Lopez, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and Modular Implicit Differentiation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5230–5242. Curran Associates, Inc., 2022. doi: 10.52202/068431-0378. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/228b9279ecf9bbafe582406850c57115-Paper-Conference.pdf.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/cbo9780511804441. URL http://dx.doi.org/10.1017/CBO9780511804441.

Hao Chen, Gonzalo E. Constante Flores, and Can Li. Physics-informed neural networks with hard linear equality constraints. *Computers & Chemical Engineering*, 189:108764, 2024.

Gonzalo E. Constante, Hao Chen, and Can Li. Enforcing Hard Linear Constraints in Deep Learning Models with Decision Rules. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 105075–105100. Curran Associates, Inc., 2025. doi: 10.52202/085713-3510. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/975a1a8e86e72c148aa2152243953064-Paper-Conference.pdf.

Carl de Boor. *A Practical Guide to Splines*. Springer New York, 1978. doi: 10.1007/978-1-4612-6333-3. URL http://dx.doi.org/10.1007/978-1-4612-6333-3.

Steven Diamond and Stephen Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016. URL http://jmlr.org/papers/v17/15-408.html.

Priya L. Donti, David Rolnick, and J. Zico Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

Rida T. Farouki. The Bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29(6):379–419, 2012. doi: 10.1016/j.cagd.2012.03.001. URL http://dx.doi.org/10.1016/j.cagd.2012.03.001.

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative Filter Networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=OmtmcPkkhT.

Paul J. Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives. *Mathematical Programming Computation*, 2026. doi: 10.1007/s12532-026-00320-7. URL http://dx.doi.org/10.1007/s12532-026-00320-7.

Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):3988–4004, 2022.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit Geometric Regularization for Learning Shapes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3789–3799. PMLR, 2020. URL https://proceedings.mlr.press/v119/gropp20a.html.

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion inbetweening. *ACM Transactions on Graphics*, 39(4), 2020.

Johannes Hendriks, Carl Jidling, Adrian Wills, and Thomas B. Schön. Linearly constrained neural networks. *arXiv preprint arXiv:2002.01600*, 2020.

R. Hettich and K. O. Kortanek. Semi-Infinite Programming: Theory, Methods, and Applications. *SIAM Review*, 35(3):380–429, 1993. doi: 10.1137/1035089. URL http://dx.doi.org/10.1137/1035089.

Prashant Khanduri, Ioannis Tsaknakis, Yihua Zhang, Jia Liu, Sijia Liu, Jiawei Zhang, and Mingyi Hong. Linearly Constrained Bilevel Optimization: A Smoothed Implicit Gradient Approach. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 16291–16325. PMLR, 2023. URL https://proceedings.mlr.press/v202/khanduri23a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/df438e5206f31600e6ae4af72f2725f1-Paper.pdf.

David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-Limited Coordinate Networks for Multiscale Scene Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16252–16262, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Lindell_BACON_Band-Limited_Coordinate_Networks_for_Multiscale_Scene_Representation_CVPR_2022_paper.html.

Lu Lu, Raphaël Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G. Johnson. Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, 2021.

Connor Magoon, Fengyu Yang, Noam Aigerman, and Shahar Kovalsky. Differentiation Through Black-Box Quadratic Programming Solvers. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 108486–108517. Curran Associates, Inc., 2025. doi: 10.52202/085713-3617. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/9bd9aa8c2326b67321eb1f5f281ba618-Paper-Conference.pdf.

Youngjae Min and Navid Azizan. Hardnet: Hard-constrained neural networks with universal approximation guarantees. *arXiv preprint arXiv:2410.10807*, 2024.

Geoffrey Négiar, Michael W. Mahoney, and Aditi S. Krishnapriyan. Learning differentiable solvers for systems with hard constraints. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vdv6CmGksr0.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer New York, 2006. doi: 10.1007/978-0-387-40065-5. URL http://dx.doi.org/10.1007/978-0-387-40065-5.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in Training PINNs: A Loss Landscape Perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 42159–42191. PMLR, 2024. URL https://proceedings.mlr.press/v235/rathore24a.html.

Abraham. Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964. doi: 10.1021/ac60214a047. URL http://dx.doi.org/10.1021/ac60214a047.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Hwijae Son, Sung Woong Cho, and Hyung Ju Hwang. Enhanced physics-informed neural networks with augmented Lagrangian relaxation method (AL-PINNs). *Neurocomputing*, 548:126424, 2023.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Runzhong Wang, Yunhao Zhang, Ziao Guo, Tianyi Chen, Xiaokang Yang, and Junchi Yan. LinSATNet: The Positive Linear Satisfiability Neural Networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 36605–36625. PMLR, 2023. URL https://proceedings.mlr.press/v202/wang23at.html.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43(5): A3055–A3081, 2021. doi: 10.1137/20m1318043. URL http://dx.doi.org/10.1137/20M1318043.

Francis Williams, Matthew Trager, Joan Bruna, and Denis Zorin. Neural Splines: Fitting 3D Surfaces With Infinitely-Wide Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9949–9958, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Williams_Neural_Splines_Fitting_3D_Surfaces_With_Infinitely-Wide_Neural_Networks_CVPR_2021_paper.html.

Francis Williams, Zan Gojcic, Sameh Khamis, Denis Zorin, Joan Bruna, Sanja Fidler, and Or Litany. Neural fields as learnable kernels for 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Jonas Zehnder, Yue Li, Stelian Coros, and Bernhard Thomaszewski. NTopo: Mesh-free topology optimization using implicit neural representations. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Fangcheng Zhong, Kyle Fogarty, Param Hanji, Tianhao Wu, Alejandro Sztrajman, Andrew Spielberg, Andrea Tagliasacchi, Petra Bosilj, and Cengiz Oztireli. Neural fields with hard constraints of arbitrary differential order. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

APPENDIX

This appendix provides: physical robot executions (Appendix A); a notation summary (Appendix A); the full training and inference algorithms (Appendix B); implementation details and a complete derivation of the differentiable projection layer (Appendix C); a scope note on nonlinear constraints (Appendix D); explicit task-specific constraint matrices for all applications (Appendix E); basis-function choices (Appendix F); extended proofs (Appendix G); and the new heat-design benchmark protocol (Appendix H).

A PHYSICAL EXPERIMENTS

We execute retargeted Unitree G1 joint trajectories on hardware as a qualitative check that the discrete retargeting outputs remain trackable. Figure 6 compares the projected source motion with onboard video frames for two short clips: turning-and-stepping and alternating steps.

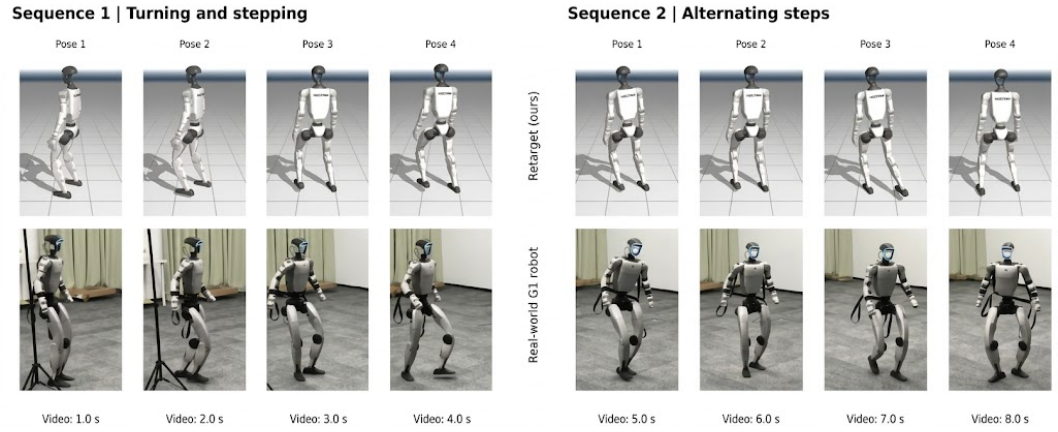


Figure 6: Comparison of retargeted motion sequences and real-world execution on the Unitree G1. Sequence 1 shows turning and stepping (video timestamps 1.0–4.0 s); Sequence 2 shows alternating steps (5.0–8.0 s). Top rows are the retargeted poses; bottom rows are real image frames.

Table 4: Summary of notation used throughout the paper.

Symbol	Meaning
$\Omega \subset \mathbb{R}^d$	spatial / temporal / spatio-temporal domain
x	query coordinate in Ω
$\Psi(x) \in \mathbb{R}^M$	basis (feature) vector; $\Psi = [\Psi_1, \ldots, \Psi_M]^\top$
$\beta \in \mathbb{R}^M$	field coefficients; $f_\beta(x) = \Psi(x)^\top \beta$
β_0	unconstrained coefficients predicted by the encoder E_θ
$\beta^\star$	projected (feasible) coefficients
$\mathcal{F}$	linear functional (point eval., trace, differential, integral)
A_{eq}, b	equality constraint matrix / right-hand side
$A_{\mathrm{lb}}, \ell;\ A_{\mathrm{ub}}, u$	lower / upper inequality blocks
G, h	stacked inequality block $G\beta \le h$
$\mathcal{C}$	feasible set $\{\beta : A\beta = b,\ G\beta \le h\}$
$\Pi_{\mathcal{C}}$	Euclidean projection onto $\mathcal{C}$
$\mathcal{A}$	active inequality set at the solution
$\tilde{A} = [A_{\mathrm{eq}}; G_{\mathcal{A}}]$	active constraint matrix
$P_{\tilde{A}}$	null-space projector, Eq. (10)
ν, λ	equality / inequality KKT multipliers

B Training and Inference Algorithms

FENCE inserts the projection layer between the encoder and any downstream objective, so training is ordinary backpropagation through a feasible point. Algorithm 1 shows the forward/backward loop. Each successfully projected output satisfies the assembled constraints to solver tolerance; the raw encoder output and the network parameters need not be feasible. Algorithm 2 shows inference: constraint assembly is amortized because A, b, G, h depend only on the basis and the task, not on the query point x.

Algorithm 1 FENCE training (one step)

Require: observations $\mathcal{O}$, target $\mathcal{L}$, functionals $\{\mathcal{F}_j\}$, basis Ψ
1: $\beta_0 \leftarrow E_\theta(\mathcal{O})$ ▷ encoder predicts unconstrained coefficients
2: assemble A_{eq}, b, G, h by stacking rows $\mathcal{F}_j[\Psi](x_i)$ ▷ Eq. (4)
3: $\beta^\star \leftarrow \Pi_{\mathcal{C}}(\beta_0)$ by solving the QP in Eq. (8) ▷ forward
4: evaluate downstream loss $\mathcal{L}(f_{\beta^\star})$ and $r = \partial\mathcal{L}/\partial\beta^\star$
5: $G_{\mathcal{A}} \leftarrow$ active rows; $\tilde{A} \leftarrow [A_{\text{eq}}; G_{\mathcal{A}}]$
6: $\partial\mathcal{L}/\partial\beta_0 \leftarrow P_{\tilde{A}}\, r$ ▷ active-set KKT backward, Eq. (11)
7: backpropagate into θ and update

Algorithm 2 FENCE inference

Require: observations $\mathcal{O}$, query points $\{x\}$
1: $\beta_0 \leftarrow E_\theta(\mathcal{O})$; assemble A_{eq}, b, G, h (cache if task-fixed)
2: $\beta^\star \leftarrow \Pi_{\mathcal{C}}(\beta_0)$ ▷ one QP solve, independent of x
3: **return** $f^\star(x) = \Psi(x)^\top \beta^\star$ for each query x ▷ feasible by construction

C Implementation and Derivation of the Projection Layer

The projection layer of Eq. (8) is implemented as a `torch.autograd.Function` (Paszke et al., 2019) (`feasible_layer.py`). The forward pass solves the convex QP $\min_\beta \frac{1}{2}\|\beta - \beta_0\|^2$ s.t. $A_{\text{eq}}\beta = b,\ G\beta \le h$ with a standard convex QP solver (OSQP). We describe the backward pass and then verify it numerically.

Derivation of the backward pass. At the optimum with active set $\mathcal{A}$, complementary slackness sets the inactive multipliers to zero, so the KKT system in Eq. (9) reduces to the equality-constrained stationarity

$$\begin{bmatrix} I & \tilde{A}^\top \\ \tilde{A} & 0 \end{bmatrix} \begin{bmatrix} \beta^\star \\ \mu \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \tilde{b} \end{bmatrix}, \qquad \tilde{A} = \begin{bmatrix} A_{\text{eq}} \\ G_{\mathcal{A}} \end{bmatrix},\ \tilde{b} = \begin{bmatrix} b \\ h_{\mathcal{A}} \end{bmatrix}, \tag{12}$$

where μ collects the active multipliers. Eliminating μ via the Schur complement of the block system (assuming $\tilde{A}$ has full row rank) gives the closed form

$$\beta^\star = \beta_0 - \tilde{A}^\top(\tilde{A}\tilde{A}^\top)^{-1}(\tilde{A}\beta_0 - \tilde{b}). \tag{13}$$

Because $\tilde{b}$ does not depend on β_0, differentiating Eq. (13) yields the symmetric, idempotent Jacobian $\partial\beta^\star/\partial\beta_0 = I - \tilde{A}^\top(\tilde{A}\tilde{A}^\top)^{-1}\tilde{A} = P_{\tilde{A}}$, i.e. the orthogonal projector onto the null space of $\tilde{A}$ (Eq. (10)). Since $P_{\tilde{A}} = P_{\tilde{A}}^\top$, the vector–Jacobian product used in reverse mode is simply $\partial\mathcal{L}/\partial\beta_0 = P_{\tilde{A}}\,(\partial\mathcal{L}/\partial\beta^\star)$, which is Eq. (11). When $\mathcal{A} = \emptyset$ we recover $\tilde{A} = A_{\text{eq}}$ and hence exactly CNF's equality projection, unifying both regimes in one formula. In practice we use the pseudo-inverse $(\tilde{A}\tilde{A}^\top)^\dagger$ to handle redundant rows on a fixed active face. At active-set transitions the projection need not be differentiable (Khanduri et al., 2023); a pseudoinverse does not remove this nonsmoothness.

Complexity. Assembly requires evaluating the chosen functionals on the basis; its cost depends on that basis and on the derivative order. After these values are cached, stacking the rows is linear in their number of nonzeros. For a dense active matrix with r rows (including equalities), forming

$\tilde{A}\tilde{A}^\top$ costs $O(r^2M)$ and factoring it costs $O(r^3)$. Applying the backward map with cached factors costs $O(rM + r^2)$ without constructing an $M \times M$ projector. Forward-solver cost also depends on sparsity, conditioning, stopping tolerance, and iterations.

Numerical verification. On randomized instances, the analytic Jacobian matches finite differences to $\max|\text{analytic} - \text{FD}| = 7.5 \times 10^{-10}$, and the returned point is feasible to machine precision: equality residual $\|A_{\text{eq}}\beta^\star - b\|_\infty \approx 2 \times 10^{-16}$ and maximum inequality violation $\max(G\beta^\star - h)_+ \approx 8 \times 10^{-16}$.

D Scope Note on Nonlinear Constraints

The experiments in this paper use linear constraints. We briefly record how nonlinear scalar limits $c_k(f) \le 0$ (e.g. von Mises stress, signed-distance collision) could in principle reuse the same inner layer, without claiming a validated algorithm. One could wrap the projection of Eq. (8) in an outer loop that linearizes each c_k around the current iterate and re-projects. Each inner solve is then exactly feasible for the assembled *linear* constraints only; convergence and feasibility for the original nonlinear constraints are not guaranteed and must be checked separately. Because a tangent to a convex inequality is generally an outer relaxation rather than an inner approximation, this route does not inherit the conservative coverage certificate that the linear constructions provide (Nocedal & Wright, 2006). We therefore treat nonlinear constraints as a scope limitation rather than a contribution, and leave a validated nonlinear solver to future work.

E Task-Specific Constraint Matrices

For a field $f_\beta(x) = \Psi(x)^\top\beta$ and a set of collocation/quadrature points, every functional becomes a matrix by stacking rows $\mathcal{F}[\Psi](x_i)$ (Eq. (4)). We first give the atomic building blocks, then the assembled blocks for each of the applications.

E.1 Atomic constraint rows

Point evaluation / interpolation (EQ). Row $\Psi(x_i)^\top$, giving $f(x_i) = \Psi(x_i)^\top\beta = g_i$.

Differential operators (EQ/INEQ). Row $\mathcal{D}\Psi(x_i)^\top$ for $\mathcal{D} \in \{\partial_k, \nabla, \Delta, \nabla\times\}$, evaluated by autodiff of the fixed basis; e.g. $\nabla f(x_i) = n_i$ yields d rows, and $\Delta f = g$ its Laplacian row.

Integral budgets (INEQ). A volume budget $\int_\Omega \rho \le V_0$ is a single row $w^\top\beta \le V_0$ with quadrature weights $w = \sum_q \omega_q \Psi(x_q)$; this is an integral *functional*, not a differential operator, which is why the framework must be stated over general linear functionals rather than differential operators only.

Box and region constraints (INEQ). Bounds $0 \le \rho(x_i) \le 1$ become $\pm\Psi(x_i)^\top\beta$ rows; a void region $\rho|_{\mathcal{V}} = 0$ is imposed as $\Psi(x_i)^\top\beta \le \epsilon$ for all collocation points $x_i \in \mathcal{V}$, which bounds the sampled interior values. With the nonnegativity rows this gives $0 \le \rho(x_i) \le \epsilon$; it does not pin unsampled locations.

E.2 Assembled blocks per case

Topology optimization. For a density field $\rho = \Psi^\top\beta$, the linear design rows can bound sampled density, enforce an interior void, and limit the quadrature volume. Elastic equilibrium $K(\rho)u = f$ is a separate state equation and is not linear in the design coefficients. The cubic construction in Appendix L uses a shared analytic void, coefficient bounds, and a volume equality. Equilibrium is solved outside the projection; stress constraints are not imposed.

Robot retargeting. The reported motion contract fixes selected joint coordinates at keyframes. If B evaluates the trajectory basis at frame times and S selects constrained coordinates, the equality block is $SB\beta = q_{\text{key}}$. Let D_1 and D_2 denote the first and second finite-difference operators at

the recorded frame rate. The inequality rows are $\pm D_1 B\beta \leq v_{\max}$ and $\pm D_2 B\beta \leq a_{\max}$, componentwise. These are discrete joint-coordinate constraints. The reported experiment focuses on joint-coordinate rate constraints; end-effector closure, collision clearance, joint-torque limits, and hardware execution are left for future work.

Scope of nonlinear extensions. Equilibrium, contact gaps, and stress limits generally depend nonlinearly on design or state. Their linearizations can be passed to the same inner layer, but require an outer algorithm and independent nonlinear validation. The experiments here focus on linear constraints; nonlinear extensions remain an active research direction.

F Basis-Function Choices

Any basis for which the required functionals $\mathcal{F}[\Psi_i]$ are well defined can be used. We consider three families. *(i) Fourier features* (Tancik et al., 2020) (Rahimi & Recht, 2007; Lindell et al., 2022) give globally smooth, band-limited Ψ whose derivatives are analytic, which is convenient for differential-operator rows. *(ii) Radial/kernel bases* (including the Gaussian kernel used by CNF (Zhong et al., 2023)) give localized influence for interpolation-heavy tasks (Williams et al., 2021). Gaussian kernels are not compactly supported, and conditioning depends on kernel width and sample placement. *(iii) Learned backbones*: the last hidden layer of a coordinate network (SIREN (Sitzmann et al., 2020) or DeepSDF-style network (Park et al., 2019)) serves as an adaptive Ψ, so the inductive bias away from collocation points is learned while feasibility is enforced in coefficient space. The band limit of Ψ controls how tightly point-wise collocation implies *continuous* feasibility between points, as discussed in the limitations.

G Extended Proofs

Theorem 1 (Feasibility at assembled constraints)**.** *Assume the feasible set $\mathcal{C}$ in Eq.* (5) *is non-empty and closed. For any network prediction $\beta_0 \in \mathbb{R}^M$, the projection in Eq.* (6) *has a unique solution $\beta^\star$. The field $f^\star(x) = \Psi(x)^\top \beta^\star$ satisfies all assembled equality and inequality functional constraints at the chosen collocation or quadrature constraints (up to QP solver tolerance).*

Proof. The objective in Eq. (6) is 1-strongly convex in β, and the feasible set is an intersection of affine half-spaces and affine equality sets, hence closed and convex. A non-empty closed convex set admits a unique Euclidean projection for every β_0 (Boyd & Vandenberghe, 2004). Since $\beta^\star \in \mathcal{C}$ by construction, it satisfies $A_{\text{eq}}\beta^\star = b$, $A_{\text{lb}}\beta^\star \geq \ell$, and $A_{\text{ub}}\beta^\star \leq u$. Each row of these matrices is obtained by applying a functional to the basis as in Eq. (4); therefore the corresponding field $f^\star$ satisfies the assembled functional constraints. □

Theorem 2 (Local differentiability and the equality-only case)**.** *Suppose the active set $\mathcal{A}$ of Eq.* (8) *is locally constant and $\tilde{A}$ has full row rank. Then $\Pi_{\mathcal{C}}$ is differentiable in a neighborhood of β_0 with Jacobian $P_{\tilde{A}}$ in Eq.* (10)*. If no inequality constraints are present, the result reduces to the equality-only CNF projection.*

Proof. With a fixed active set, the KKT system reduces locally to the equality-constrained projection

$$\min_{\beta} \frac{1}{2}\|\beta - \beta_0\|^2 \quad \text{s.t.} \quad \tilde{A}\beta = \tilde{b}. \tag{14}$$

Solving the equality KKT system gives

$$\beta^\star = \beta_0 - \tilde{A}^\top(\tilde{A}\tilde{A}^\top)^{-1}(\tilde{A}\beta_0 - \tilde{b}). \tag{15}$$

Differentiating with respect to β_0 yields Eq. (10). If there are no inequalities, $\tilde{A} = A_{\text{eq}}$, which is exactly the equality projection used by CNF. Thus the equality-only coefficient projection is recovered when the inequality block is absent. □

Corollary 1 (Projection and coefficient optimization)**.** *For a fixed nonempty closed convex set $\mathcal{C}$, a coefficient update $\beta^{k+1} = \Pi_{\mathcal{C}}(\beta^k - \eta\nabla_\beta L(\beta^k))$ is projected gradient descent on $\mathcal{C}$.*

Proof. The update is the Euclidean projection of a gradient step, which is the definition of projected gradient descent. Existence and uniqueness follow from the nonempty closed convex feasible set. □

Training an encoder through $L(\Pi_{\mathcal{C}}(E_\theta(\mathcal{O})))$ instead updates θ by the chain rule. While this is not projected gradient descent in coefficient space, it efficiently propagates gradients through the projection layer, and the projected output remains feasible by construction.

Proposition 1 (Non-expansiveness / stability)**.** *For a fixed nonempty closed convex set $\mathcal{C}$, the projection $\Pi_{\mathcal{C}}$ is 1-Lipschitz (firmly non-expansive): $\|\Pi_{\mathcal{C}}(\beta_0) - \Pi_{\mathcal{C}}(\beta_0')\| \le \|\beta_0 - \beta_0'\|$ for all β_0, β_0'. Consequently a bounded perturbation of the encoder output produces an at-most-equally-bounded change of the projected coefficients. At a query x, the corresponding field perturbation is bounded by $\|\Psi(x)\|_2\|\beta_0 - \beta_0'\|_2$. This guarantees stability under bounded perturbations for a fixed feasible set and Euclidean metric.*

Proof. Firm non-expansiveness of the Euclidean projection onto a non-empty closed convex set is standard: for $p = \Pi_{\mathcal{C}}(\beta_0)$, $p' = \Pi_{\mathcal{C}}(\beta_0')$, the variational inequalities $\langle \beta_0 - p,\, z - p\rangle \le 0$ and $\langle \beta_0' - p',\, z - p'\rangle \le 0$ hold for all $z \in \mathcal{C}$; taking $z = p'$ and $z = p$ respectively and adding gives $\|p - p'\|^2 \le \langle \beta_0 - \beta_0',\, p - p'\rangle \le \|\beta_0 - \beta_0'\|\,\|p - p'\|$, hence $\|p - p'\| \le \|\beta_0 - \beta_0'\|$. □

The null-space Jacobian is also derived in Appendix C, Eq. (13).

H Reproducible Protocol for Inverse Heat-Source Design

This appendix documents the original sparse benchmark. Its rows are assembled at sampled points only, so the FENCE variant evaluated here is reported throughout as *FENCE with sampled constraints*, in contrast to the sufficient construction of Appendix I.

Representation and objective. For $m, n \in \{1, 2, 3, 4\}$, define $\psi_{mn}(x, y) = \sin(m\pi x)\sin(n\pi y)$ and $\lambda_{mn} = 0.03\pi^2(m^2 + n^2)$. We represent $q = \sum_{mn} z^q_{mn}\psi_{mn}$ and $T = \sum_{mn} z^T_{mn}\psi_{mn}/\lambda_{mn}$. The PDE is equivalent to $z^T = z^q$; the Dirichlet boundary holds by construction. Projecting the paired coefficient vector onto this equality averages its two halves. Subsequent Euclidean projection in the common 16-dimensional coefficient vector is equivalent to joint projection onto the full mixed set, because the reduced metric is a scalar multiple of the identity. The encoder has two width-64 Tanh hidden layers and a 32-dimensional linear output. The decoder is fixed; only the encoder is trained.

The task loss is $J = \mathrm{mean}_{X_{16}}(T - T_d)^2 + 10^{-4}\,\mathrm{mean}_{X_{16}}\, q^2$, where X_{16} is the 16×16 interior grid and $T_d = a\exp(-\|x - c\|^2/(2s^2))$. The 9 input parameters are c_x, c_y, a, s, protected-disk center p_x, p_y, its radius r, its temperature cap, and the power budget. Independent uniform training/ID ranges are $c_x, c_y \in [0.22, 0.78]$, $a \in [0.30, 0.65]$, $s \in [0.12, 0.23]$, $p_x, p_y \in [0.20, 0.80]$, $r \in [0.09, 0.16]$, $T_{\mathrm{safe}} \in [0.10, 0.24]$, and $P_{\max} \in [0.30, 0.65]$. The shifted test changes only the last two ranges to $[0.055, 0.095]$ and $[0.20, 0.29]$. NumPy generation seeds are 100/200/300/400 for train/validation/ID/shift. A zero source and zero temperature are always feasible.

Constraint assembly and solver details. Source bounds use 64 interior points; the disk uses its center plus eight equally spaced points on each of the half-radius and full-radius rings. The source integral is analytic. All methods receive identical rows. OptNet uses official qpth 0.0.18 in float64, tolerance 10^{-10} and at most 60 iterations. FENCE uses Clarabel (Goulart & Chen, 2026) for the forward QP (requested gap/feasibility tolerances 10^{-11}, at most 300 iterations), followed by the active-face null-space Jacobian. This is an implementation of the coefficient projection. Constraint parameters are fixed inputs during training, so differentiation is needed with respect to predicted coefficients, not learned constraint matrices.

The CNF-style baseline uses the common exact equality projection plus a training penalty on squared inequality residuals. DC3 uses the same equality completion, 10 differentiable inequality-correction steps in training, and up to 1,000 in evaluation, stopping below 10^{-7}. Its training penalty uses the residual ℓ_2 norm as in the authors' implementation. Reduced correction minimizes squared positive residuals with step $0.9/(2\|G\|_2^2)$ and zero momentum. These are controlled

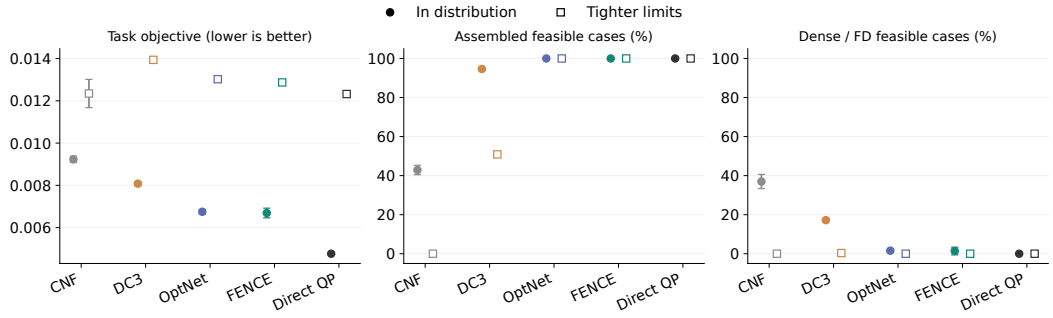


Figure 7: **Heat-control objective and feasibility across test regimes.** Filled circles: in-distribution; open squares: tighter limits. Error bars are one standard deviation across three training seeds; Direct QP is deterministic. Dense feasibility checks use independent finite-difference temperatures and the full-grid source, not only the assembled rows.

Table 5: **Controlled inverse heating with sampled constraints.** Task objective J is lower-is-better; mean $\pm$ standard deviation across three training seeds. FENCE here denotes the sampled-constraint variant of Appendix H; it is not the sufficient construction of Table 3. Feasible-case percentages use tolerance 10^{-6}; paired entries are in-distribution / tighter-limit tests. Dense feasibility requires source bounds, power budget, and the independently recomputed temperature cap. An infeasible method's smaller objective is not a better feasible solution.

Method	$10^3 J_{\mathrm{ID}}$	$10^3 J_{\mathrm{tight}}$	Assembled (%)	Dense/FD (%)
CNF-style + penalty	9.231 ± 0.149	12.345 ± 0.668	42.9 / 0.0	37.0 / 0.0
DC3 (adapted)	8.081 ± 0.091	13.939 ± 0.064	94.6 / 50.9	17.2 / 0.3
OptNet	$\underline{6.750} \pm \underline{0.106}$	$\underline{13.023} \pm \underline{0.124}$	**100.0 / 100.0**	1.6 / 0.0
FENCE	$\mathbf{6.692 \pm 0.229}$	$\mathbf{12.871 \pm 0.076}$	**100.0 / 100.0**	1.4 / 0.0
Direct task QP	4.765	12.322	100.0 / 100.0	0.0 / 0.0

task adaptations, not claims to reproduce the original papers' full models. For each method, two 200-update seed-0 validation runs select either penalty weight $\{0.1, 1\}$ at learning rate 10^{-3} (CNF-style/DC3), or learning rate $\{10^{-3}, 3 \times 10^{-3}\}$ (OptNet/FENCE). The validation score is mean $J + 10\|[Gz - h]_+\|_\infty^2$. Final runs use 800 updates and seeds 0, 1, 2 with identical initializations and minibatch orders; gradient norms are clipped at 5. No optimal-QP labels are supplied during training.

Independent evaluation and numerical checks. Dense source evaluation and protected-disk masks use all 96×96 interior points. Temperatures are recomputed from the predicted source with the five-point finite-difference Poisson operator, using its discrete sine eigenvalues rather than the continuum eigenvalues. This implementation agrees with a sparse direct solve to below 10^{-12} in the numerical check. Projection directional derivatives are checked by central finite differences; the recorded absolute discrepancy is 3.3×10^{-6} on the fixed validation probe. For the first 32 held-out FENCE predictions, the identical-input OptNet/FENCE coefficient discrepancy is 5.06×10^{-9}. The raw per-case arrays, checkpoints, training histories, selected configurations, package versions, and file hashes are saved with the experiment. Examples 0, 1, and 2 were selected before training; the main text shows example 0.

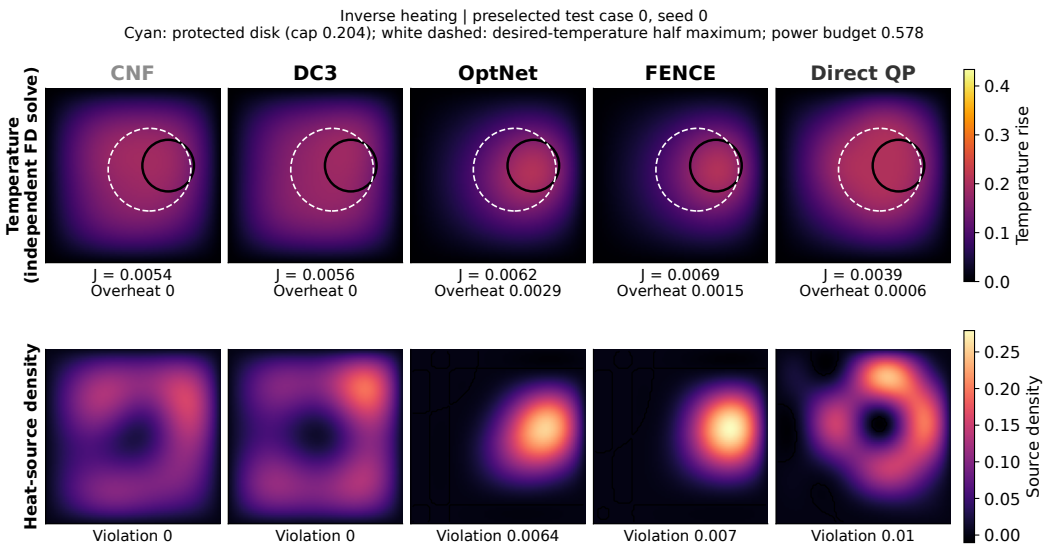


Figure 8: **Heat-source design with a protected region under sampled constraints.** Predeclared test case 0, seed 0; no method-dependent example selection. Top: temperatures recomputed with the independent finite-difference solver; cyan marks the protected disk and white dashes the desired profile's half-maximum contour. Bottom: sources, with contours marking the dense-grid source regions outside the bound. Color scales are shared across methods. "Overheat" and "Violation" report maximum dense-grid residuals between assembled points, which the sufficient construction of Appendix I removes; the render is a numerical field visualization.

I Sufficient Heat Constraints and Frozen Evaluation

The sparse-row failure in Table 5 motivates a change to constraint compilation. We keep the 16-mode sine representation, but replace sampled source bounds with sufficient coefficient bounds and add a coverage margin to the protected-region temperature rows. The frozen experiment below trains new encoders and evaluates independent tasks under these sufficient constraints. It is separate from the original sparse benchmark.

A sufficient source bound. Using $\sin(m\pi x) = \sin(\pi x)U_{m-1}(\cos \pi x)$, write

$$q(x, y) = \sin(\pi x) \sin(\pi y) \sum_{a,b=0}^{3} \gamma_{ab} B_a^3(u) B_b^3(v), \qquad (u, v) = \tfrac{1}{2}(1 + \cos \pi x, 1 + \cos \pi y),$$

where B_a^3 are cubic Bernstein polynomials (Farouki, 2012) and U_{m-1} are Chebyshev polynomials of the second kind. A fixed invertible linear transformation relates the sine coefficients to these coefficients: $z = C\gamma$. Nonnegativity and partition of unity of the Bernstein basis imply that $0 \le \gamma \le 2$ is sufficient for $0 \le q(x, y) \le 2$ throughout the unit square. This condition is sufficient rather than necessary and thus changes the feasible set.

A sufficient temperature bound. For the continuum Poisson temperature, the triangle inequality gives

$$\|\nabla T(x, y)\|_2 \le \sum_{mn} \frac{\pi\sqrt{m^2 + n^2}}{\lambda_{mn}} |z_{mn}| \le w^\top \gamma, \quad w^\top = \left[\frac{\pi\sqrt{m^2 + n^2}}{\lambda_{mn}}\right]_{mn}^\top |C|.$$

The last bound uses $\gamma \ge 0$. A polar cover of a disk of radius r, with 64 radial intervals and 256 angular samples, has covering radius at most $\delta = r(1/128 + \pi/256)$. At each cover node x_i, the affine row

$$\left[\psi(x_i)^\top \operatorname{diag}(\lambda)^{-1} C + \delta w^\top\right] \gamma \le T_{\text{safe}}$$

therefore suffices for the temperature cap throughout the disk. We retain the analytic power row. Constraint exchange adds the eight most violated cover rows per iteration and terminates only after

scanning all cover nodes with residual at most 10^{-8}, with a 61-iteration cap. The projection and direct optimum share the union of exchanged rows. Bounds concern the continuum sine field; the independent finite-difference solution is checked separately.

Projection and task optimization. Projection minimizes $\frac{1}{2}\|C\gamma - z_0\|^2$; Direct QP minimizes the task objective under the same sufficient rows. qpth independently solves the identical projection. An initial twelve-validation-task diagnostic showed that replacing infeasible sampled bounds with sufficient rows increases the mean projection objective by 31.9%, reflecting the cost of guaranteed continuous coverage. The coverage–quality sweep in Appendix N quantifies this conservatism with fixed source bounds and multiple temperature covers.

Frozen three-seed extension. We use 128 training tasks (generation seed 5100), 32 validation tasks (5200), 200 Adam updates (Kingma & Ba, 2015), batch size eight, and learning rate 10^{-3} for every method. Four configurations per method are compared with paired initialization seed 10: penalty weights $\{0.01, 0.1, 1, 10\}$ for the CNF-style and DC3 adaptations, and projection-distance weights $\{0, 0.001, 0.01, 0.1\}$ for both QP methods. The latter adds $\rho\|z_0 - \mathrm{stopgrad}(z^\star)\|^2/16$ to the training loss. Selection first maximizes validation dense/FD feasibility and then minimizes the task objective. The chosen values are 10 and 0.1, respectively. We freeze these settings, train paired seeds 11, 12, and 13, and evaluate all models on the same 96 new tasks (5300). There is no subsequent test-set selection. All runs use two CPU threads; each training run has a 240-second ceiling, and all finish within it.

Rows are divided by their Euclidean norm and by the square root of their group's cardinality (source lower, source upper, power, temperature). This balances the residual losses without changing the feasible set. Both exact QP solvers receive the same exchanged rows, renormalized to unit norm for numerical conditioning. All reported certificate residuals are restored to physical units and checked over the full row set. DC3 uses ten correction steps during training and at most 1,000 at evaluation, as in the development protocol.

Both QP implementations pass all 288 dense-grid and independent FD checks, as does the direct optimum on its 96 tasks. Failing a sufficient row does not necessarily imply a physical violation, which explains DC3's different rates under these two checks. FENCE and OptNet report comparable mean J in Table 3 (8.463 ± 0.195 vs. 8.507 ± 0.276), while sharing 100% sufficient-row and dense/FD coverage; the direct optimizer attains a lower objective (6.774) with full task information.

Numerical derivative audit. Positive row rescaling initially exposed a threshold-sensitive active-face implementation: scaled slacks misclassified inactive rows and produced spurious zero gradients. That run was stopped and excluded from selection. Canonical row normalization, a 10^{-13} forward-solve tolerance, and rank-aware active-face differentiation repair the fixed development probes. Five recorded directional checks have maximum absolute AD–finite-difference discrepancy below 1.9×10^{-6} at step 10^{-6}. One probe crosses an active-face boundary at step 10^{-4} and shows a larger discrepancy, as expected near nonsmooth transitions. Local checks validate the Jacobian where the active set is stable.

J Discrete Retargeting: Protocol and Independent Evaluation

Four predeclared public motion excerpts. We use `walk1_subject1`, `dance1_subject2`, `run1_subject2`, and `fight1_subject2` from LAFAN1. Each is processed by the official GMR BVH-to-G1 configuration, with the optional velocity limit disabled and DAQP as the IK solver. The first 60 source frames warm up IK; source frames 60–209 form the evaluated 150-frame excerpt at 30 Hz. Excerpts were named before running these comparisons, and all four are reported. This is a correction-stage pilot, not LAFAN1's motion-in-betweening protocol. The 29 joint samples are the coefficients. Upper-body joints 15–28 are fixed at local frames $\{0, 37, 74, 112, 149\}$; first and second differences act on all joints. We minimize $\frac{1}{2}\|q - q_0\|_F^2$ with OSQP tolerances 10^{-9} and polishing. Every QP reports solved. Counts use the declared limits plus 10^{-5}.

The same fixed filter (17-sample Savitzky–Golay, degree three) (Savitzky & Golay, 1964) and penalty settings described below are applied to every excerpt. Their mean RMS displacements are 1.684° and 1.438°, compared with FENCE's 1.782°. While filtering and penalty methods achieve

Table 6: **Four-motion correction pilot.** All FENCE keyframe RMSEs are below 10^{-12} degrees and both rate-violation counts are zero. Motion averages weight each excerpt equally; these are distinct motions rather than different frames of one clip.

Motion	Raw peak speed rad/s	Raw peak acceleration rad/s^2	FENCE RMS change degrees	Feasible
Walking	7.397	101.076	2.084	yes
Dancing	6.055	132.116	0.365	yes
Running	8.112	172.938	1.065	yes
Fighting	11.822	514.342	3.616	yes

smaller displacements, they do not produce feasible solutions: mean keyframe RMSE is $0.926°$ for filtering and $0.00333°$ for the penalty, yet both leave rate violations. The fixed penalty's mean acceleration-violation fraction increases from 13.73% to 14.86%, demonstrating that soft constraints do not ensure feasibility even with large penalty weights.

GMR's optional velocity limits. We rerun all four excerpts with GMR's native optional IK velocity limit (3π rad/s), and with an explicitly adapted internal limit of 2.5 rad/s. Raw keyframes and evaluation thresholds remain unchanged. In the checked-out GMR revision, the limit list is passed as the sixth positional argument to `mink.solve_ik`; with Mink 1.3.0 that position is `safety_break`. An evaluation adapter forwards `limits` by keyword without changing the IK objective. Rerunning the no-velocity-limit configuration reproduces every stored raw joint value exactly. The earlier, uncorrected runs did not activate velocity limits and are excluded from the comparison.

Table 7: **GMR limit controls on the same four pilot excerpts.** Equal-weight averages across motions. Anchor RMSE and RMS change are in degrees; violations use the common offline limits 2.5 rad/s and 20 rad/s^2. The IK options do not enforce the offline keyframes.

Control	Anchor RMSE	RMS change	Speed viol. (%)	Accel. viol. (%)
Raw GMR	0	0	4.33	13.73
GMR native limit	0.470	2.626	3.76	16.33
GMR adapted limit 2.5	7.263	7.582	0.49	15.53
FENCE correction	$< \mathbf{10^{-12}}$	**1.782**	**0**	**0**

The native limit constrains each internal IK integration step ($\Delta t = 0.002$ s). Multiple steps are accumulated before emitting a frame at 30 Hz, so setting its internal limit to 2.5 does not itself bound the emitted finite differences by 2.5. Nor does it constrain second differences. These options and the offline correction address different contracts; comparisons with the complete GMR pipeline or time-matched online controllers are left for future work.

J.1 Frozen Subject-Held-Out Correction

The four pilot excerpts above serve only as development data for this additional protocol. Before loading new trajectories, we fix ten excerpts from subjects 3, 4, and 5: for each subject and each exact action prefix (walk, dance, run, fight), take the lexicographically first available BVH. Unavailable action–subject pairs are omitted without replacement. The resulting ordered list is `walk2_subject3`, `dance1_subject3`, `fight1_subject3`, `walk2_subject4`, `dance2_subject4`, `run2_subject4`, `walk1_subject5`, `dance2_subject5`, `run1_subject5`, and `fight1_subject5`. The same 60-frame IK warm-up, 150-frame excerpt, raw anchors, rate limits, and G1 model are used throughout. Subjects are distinct from development subjects 1–2; multiple excerpts from a subject are not treated as independent subjects for significance testing.

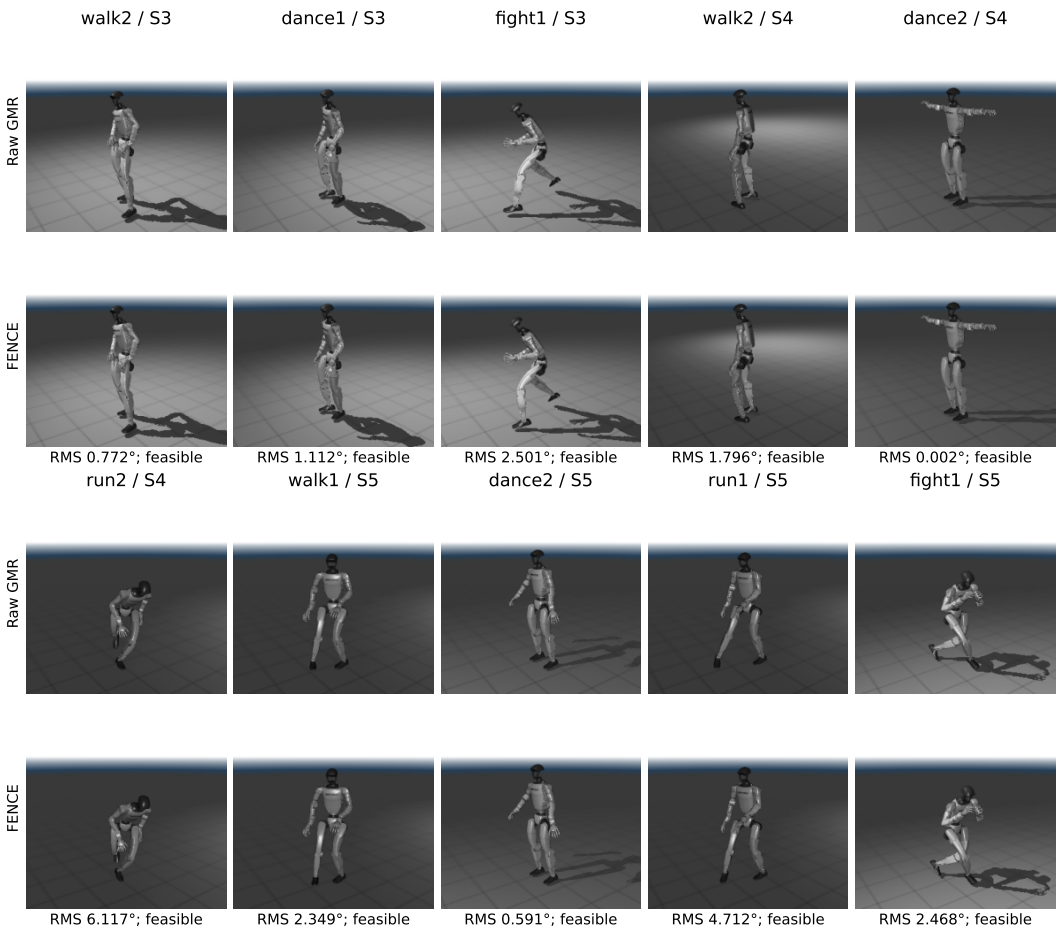


Figure 9: **All ten frozen motion excerpts.** Each raw/FENCE pair uses the same frame (local time 2.5 s), camera, and root pose. Labels give full-excerpt RMS joint correction; all ten FENCE trajectories meet the discrete contract. Poses are MuJoCo forward kinematics of the saved evaluated arrays.

Controls and frozen selection. Let Dq stack first differences divided by $2.5/30$ and second differences divided by $20/30^2$. In addition to raw GMR/equality-only and the hard projection, we evaluate

$$\min_{q,s\geq 0} \ \tfrac{1}{2}\|q-q_0\|_F^2 + \tfrac{\lambda}{2}\|s\|_2^2 \quad \text{s.t.} \quad Eq = Eq_0, \qquad -\mathbf{1} - s \leq Dq \leq \mathbf{1} + s.$$

This convex soft-penalty problem is solved with Clarabel; each joint is separable. The filter control applies a degree-three Savitzky–Golay filter and then restores exactly the selected raw anchors, so its equalities are not disadvantaged. Penalty candidates are $\lambda \in \{0.01, 1, 100, 10000\}$; filter windows are $\{9, 17, 33, 65\}$. Each receives four development candidates. Selection first maximizes fully feasible development excerpts, then minimizes mean normalized maximum rate excess, then squared correction. This selects $\lambda = 10000$ and window 9; settings are recorded before any held-out BVH is loaded. The hard QP has no task hyperparameter selected from these trajectories.

FENCE uses OSQP with absolute/relative tolerances 10^{-10}; the independent hard-QP reference uses Clarabel with feasibility and gap tolerances 10^{-10}. Each method has a 120-second per-excerpt ceiling, with four seconds per joint; no ceiling is reached. Both hard solvers and every soft solve report success. Hard-solver outputs differ by at most 2.03×10^{-5} radians, with mean RMS corrections agreeing to 1.2×10^{-9} degrees. No encoder is trained here, so training-seed averages are inapplicable; uncertainty from subject, embodiment, and scene variation remains distinct from numerical solver agreement.

The selected penalty uses the largest tested weight. Even at this extreme setting, small residual violations persist, demonstrating that penalty methods cannot guarantee feasibility regardless of weight

Table 8: **Ten frozen excerpts from three held-out subjects.** RMS change is averaged equally over excerpts. Excesses are maxima over all entries and excerpts, in rad/s and rad/s^2. Full feasibility requires anchor error $\leq 10^{-8}$ rad and rate excess $\leq 10^{-5}$. The selected penalty nearly matches the hard optimum despite failing this strict tolerance.

Method	Feasible	RMS change (°)	Speed excess	Accel. excess
Raw GMR / equality-only	0/10	0	13.457	508.892
Filter + equality	1/10	<u>0.64270</u>	13.796	251.599
Soft QP, $\lambda = 10^4$	1/10	2.24203	1.01×10^{-4}	3.58×10^{-4}
FENCE	**10/10**	**2.24205**	$\mathbf{< 10^{-10}}$	$\mathbf{< 10^{-10}}$
Direct task QP	10/10	2.24205	$< 10^{-10}$	$< 10^{-9}$

magnitude. This result highlights the fundamental advantage of explicit constraint enforcement for safety-critical applications.

K Support-Based Coverage and Representation Cost

This diagnostic separates continuous void exclusion from topology optimization. It uses the three void geometries specified in Appendix L and nonnegative piecewise-linear tensor-product hats $\psi_{ij} = \phi_i\phi_j$. For a closed rectangular support R_{ij} and an ellipse $V = \{x : \sum_{k=1}^{2}((x_k - c_k)/r_k)^2 \leq 1\}$, define

$$d_{ij}^2 = \sum_{k=1}^{2}\left(\frac{\mathrm{clip}(c_k, R_{ij,k}) - c_k}{r_k}\right)^2.$$

Setting $\beta_{ij} = 0$ whenever $d_{ij}^2 \leq 1$ (with a 10^{-12} numerical margin) removes every support intersecting the void. With $0 \leq \beta \leq 1$, nonnegativity and partition of unity imply $0 \leq \rho \leq 1$ throughout the domain and $\rho|_V = 0$. This is conservative: it removes usable material outside the void as well.

Capacity and exact integrals. For $C_x \times C_y$ uniform hat cells, normalized one-dimensional integrals are $w_i^x = 1/C_x$ at interior nodes and $1/(2C_x)$ at endpoints, with the analogous w_j^y. The exact volume row is $\sum_{ij} w_i^x w_j^y \beta_{ij} = 0.4$ after support elimination. Within a nonnegative representation, if $\psi_j(x) > 0$ anywhere in V, then $0 \leq \beta_j\psi_j(x) \leq \rho(x) = 0$ requires $\beta_j = 0$. Eliminating such functions is therefore necessary within this representation as well as sufficient. The closed-support rule can additionally remove boundary-tangent functions; no such tangency is detected within the stated threshold in these instances. This argument does not apply to signed representations with cancellation.

Set all remaining coefficients to one. Their integral sum c is the maximum material capacity, compared with $c_\infty = 1-|V|/|\Omega|$ for an unrestricted density. Across all three geometries, the predeclared bases 16×6, 32×12, 64×24, and 128×48 have mean capacity gaps 22.32%, 10.64%, 4.98%, and 2.51% of domain volume. These are representation costs, not compliance losses. Two-point Gauss integration per basis cell agrees with the analytic capacities within 2.23×10^{-15}.

Coverage and useful design are separate requirements. A comparison assembly removes only hats positive at void samples on a fixed 64×24 cell-center grid. Both assemblies share exact volume and coefficient bounds. Setting every free coefficient to $0.4/c$ gives a feasible volume-and-box probe in all 24 configurations. Evaluation uses the same 100,000 interior points, 4,096 boundary points per geometry, and 256×96 and 512×192 grids. All support-excluded probes have zero measured void density. All sampled probes satisfy their assembled rows but leak between them; at 128×48, maximum void densities are 0.423, 0.414, and 0.417, demonstrating the necessity of continuous coverage guarantees. The continuous exclusion follows from the support argument, not from these finite probes.

Sufficient coverage alone does not ensure a useful mechanical design. At 16×6, support exclusion disconnects the tip-case load path; a completed preliminary 500-step setting has compliance about 1.26×10^7 for every method. Separately, a sampled-support topology run can retain density 0.1473

inside the analytic void on a finer grid, despite zero assembled violation. These observations motivate the distinct cut-basis representation below, which combines continuous coverage with practical design quality.

L CUBIC TOPOLOGY: REPRESENTATION, CONTROLS, AND EVALUATION

This appendix specifies the three-case experiment in Figure 5. The domain is $[0, 1.5] \times [0, 0.5]$. Plane-stress elasticity uses $E = 1$, $\nu = 0.3$, and $E_e = 10^{-6} + (1 - 10^{-6})\rho_e^3$. Cantilever cases clamp the left edge and apply a unit downward load at the lower-right (tip) or middle-right (shear) node. Half-MBB fixes horizontal displacement on the left edge and vertical displacement at the lower-right node, with a unit downward load at the upper-left node. Tip and shear voids are ellipses centered at $(0.75, 0.25)$ and $(0.70, 0.28)$, with radii $(0.30, 0.17)$ and $(0.26, 0.15)$; the half-MBB void is a circle centered at $(0.48, 0.32)$ with radius 0.18. Mechanical evaluation uses $\mathrm{clip}(\rho, 0, 1)$ for all methods, while feasibility is measured before clamping. Infeasible designs cannot be ranked by compliance alone because this extension can change their physical volume.

Representation and admissibility. Let $B_i^3(x)B_j^3(y)$ be open-uniform cubic tensor-product B-splines on $[0, 1.5] \times [0, 0.5]$, with 64×24 knot spans and 67×27 coefficients before removing zero-integral functions. For the same elliptical void V, define

$$\widetilde{\psi}_{ij}(x, y) = \mathbf{1}_{\Omega \setminus V}(x, y)B_i^3(x)B_j^3(y), \qquad \rho = \sum_{ij} \widetilde{\psi}_{ij}\beta_{ij}.$$

All methods receive this identical cut basis. Its nonnegativity and partition of unity outside V imply that $0 \le \beta \le 1$ suffices for $0 \le \rho \le 1$, while $\rho|_V = 0$ holds by construction. The field is C^2 away from the void boundary and may jump across that boundary. This is a continuous density representation, not a binary manufactured design. The common volume row is $w^\top \beta = 0.4$, where $w_{ij} = |\Omega|^{-1} \int_{\Omega \setminus V} B_i^3 B_j^3$. B-spline antiderivatives integrate rectangles; adaptive quadrature subtracts their elliptical intersections. Reassembling whole-basis integrals at a stricter tolerance changes individual weights by at most 6.66×10^{-17}. Quadrature error estimates provide numerical volume accuracy; analytic hat integrals in Appendix K offer exact certificates.

Shared NTopo-inspired updates. We inspect the public NTopo implementation[1] and adapt its sinusoidal proposal architecture, density-weighted sensitivity filter, and OC-target fitting (Zehnder et al., 2021). Here the network predicts coefficients, whereas NTopo predicts pointwise density and uses a learned displacement field. We retain 192×72 Q4 finite elements, cell-average density, and the SIMP constitutive model specified above.

All three neural methods use width 60, five sine hidden layers with skip connections, first-layer frequency 60, and a sigmoid output. A zero final layer gives the same initial constant density. At each of 40 outer steps, we filter physical density sensitivities with a cone radius $2(1.5/192)$ and denominator floor 10^{-3}, pull them back to coefficients, and construct an OC target with move limit 0.2 and damping 0.5. The target generator first projects its base coefficients to the common feasible set for *every* method. Each neural method fits its own target using five Adam steps at rate 10^{-3}, an integral-weighted coefficient loss, and the same squared bound penalty of weight 10. FENCE uses mixed projection at the network output; the CNF-style control projects the volume equality only. Direct OC applies the same target update without a network. Thus OC shares 40 physical updates but has no 200-step network-fitting cost.

All three predeclared development geometries use seed 41. Every method completes 40 outer updates within the same 2400-second per-method ceiling.

Explicit OptNet precision variant. Both QP methods minimize the same Euclidean projection objective with the same volume and box rows. Default qpth numerical settings did not meet the predeclared 10^{-5} forward/VJP/directional-FD checks on some inputs, including a trained proposal. We therefore developed a precision variant with tightened tolerances that passes all validation probes. We retain these failed probes and create a separately named variant of qpth 0.0.18: its CVXPY forward calls OSQP (Diamond & Boyd, 2016) with absolute and relative tolerances 10^{-11}, polishing,

[1] `https://github.com/JonasZehn/ntopo`, commit `d3e17ca4`.

and at most 100,000 iterations; its KKT backward slack/dual floor changes from 10^{-8} to 10^{-10}. The objective is multiplied by 10^4, inequality rows by 100, and the volume row is normalized. These positive scalings preserve the feasible set and exact minimizer. The variant retains qpth's KKT derivative; it does not substitute the FENCE derivative. Seven constant, random, and trained-snapshot inputs pass the original thresholds; the snapshot VJP maximum difference is 3.57×10^{-6}. Original sources, the MIT license, variant hashes, and all failed probes are retained. This finite numerical check does not prove derivative accuracy at every degenerate active set.

Table 9: **Cubic development results, seed 41.** Compliance / maximum coefficient-bound violation, before clamping. Both QP controls and OC are feasible within 10^{-6}; CNF-style entries record the residual violations left by its penalty. Infeasible designs are not ranked by compliance alone.

Method	Tip	Half-MBB	Shear
CNF-style + penalty	299.37 / .00282	1738.53 / .00741	268.12 / .00955
OptNet (precision variant)	299.38 / 4.3×10^{-12}	1706.15 / 0	264.97 / 0
FENCE	296.68 / 0	1704.77 / 0	265.08 / 0
Direct OC	292.50 / 0	1726.90 / 0	260.09 / 0

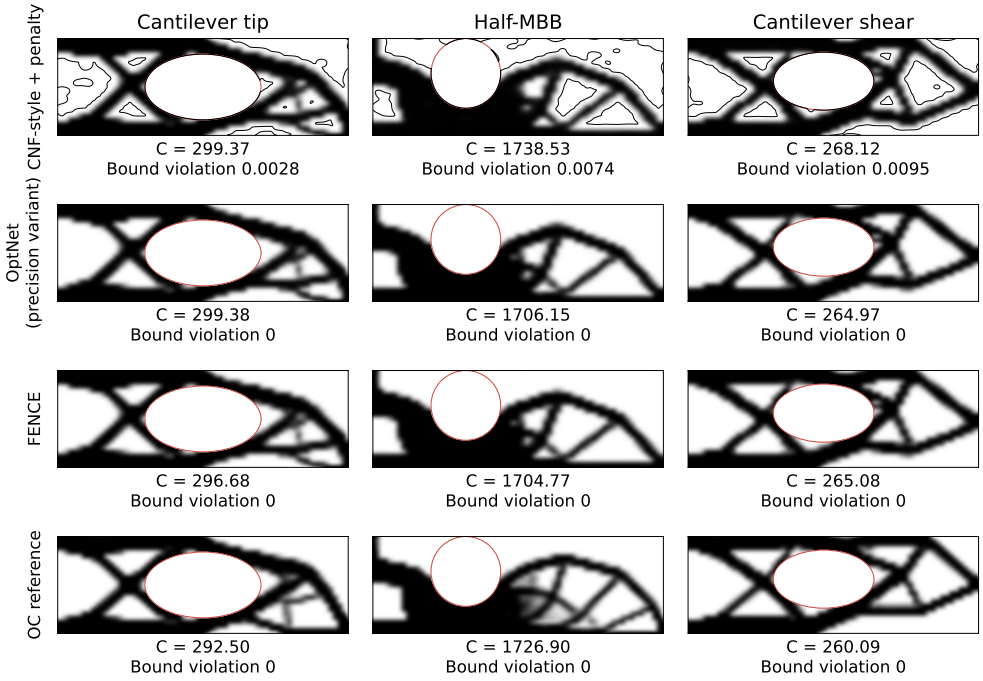


Figure 10: Complete cubic-field comparison. All four methods on all three development geometries at seed 41: CNF-style + penalty, OptNet (precision variant), FENCE, and Direct OC. The CNF-style and FENCE rows are also shown in the focused main-text view (Figure 5). Each panel evaluates the saved field at 1536×512 points with shared gray scales and the same analytic void cut (red). Labels report compliance C and maximum field-bound violation before the common $[0, 1]$ mechanical extension. The QP controls satisfy the bounds while the penalized control leaves residual violations. Direct OC reaches lower compliance on tip and shear, while the QP controls are competitive on half-MBB.

Frozen-field checks and interpretation. All twelve fields are evaluated without further optimization on FEM grids 192×72, 256×96, and 320×120, using independently assembled NumPy/SciPy mechanics under the same Q4/SIMP model. New cell arrays reproduce the training-grid fields within 3.78×10^{-15}; equilibrium and energy-identity residuals pass 10^{-7}. Every field has zero density at 100,000 new random void points and 4,096 boundary points. Whole-basis volume residuals are at most 1.01×10^{-14}. CNF-style maximum field-bound violations on the 1536×512 display grid are 0.00277, 0.00736, and 0.00946; the QP and OC fields satisfy the sufficient coefficient bounds within tolerance. FENCE's final two FEM levels differ by 0.131%, 0.148%, and 0.155% in compli-

ance. Across all methods, this change is at most 0.159%. These are refinement diagnostics, not a continuum convergence theorem.

FENCE and the OptNet precision variant achieve comparable compliance on tip, half-MBB, and shear for seed 41 (Table 9); Direct OC attains lower compliance on tip and shear. Paired initialization results below quantify consistency across seeds.

Recorded local two-thread times are 24.4–24.8 seconds for FENCE, 201.8–228.5 for the OptNet precision variant, 17.9–18.6 for CNF-style, and 17.1–17.8 for OC. These are actual implementation costs, not an architecture-matched speed theorem or a comparison with the original qpth default backend.

Paired initialization checks. Two further seeds, 42 and 43, were fixed before observing their outcomes. All other settings, preflight thresholds, and per-method budgets remain unchanged. All 18 additional neural settings complete; OC is the same deterministic reference and is not counted as three independent samples. Table 10 reports all seeds, rather than selecting a favorable initialization. Figures 5 and 10 retain the original seed 41. These results concern equivalent projection implementations on the three development geometries, not comparisons with complete topology-optimization systems.

Table 10: Cubic projection-layer control over seeds 41–43. Mean compliance $\pm$ sample standard deviation. Parentheses report complete-contract passes out of three; infeasible CNF-style results are diagnostic and cannot be ranked as feasible solutions. OC is a single deterministic reference.

Method	Tip	Half-MBB	Shear
FENCE	294.77 ± 2.28 (3/3)	1719.02 ± 17.71 (3/3)	265.42 ± 2.08 (3/3)
OptNet (precision variant)	295.76 ± 3.58 (3/3)	1717.27 ± 12.76 (3/3)	265.53 ± 2.58 (3/3)
CNF-style + penalty	297.32 ± 1.79 (0/3)	1727.31 ± 11.01 (0/3)	267.33 ± 3.02 (1/3)
Direct OC	292.50	1726.90	260.09

The new fields pass independent training-grid mechanical replay, reassembled volume, and random void/boundary checks. Sufficient coefficient bounds hold in all nine FENCE and nine OptNet runs; CNF-style passes only one of nine complete contracts. Its new fields are additionally sampled on a 768×256 grid. These checks do not repeat or extend the three-grid refinement claim beyond seed 41. FENCE and OptNet have similar mean compliance (Table 10): FENCE is 0.33% lower on tip and 0.04% lower on shear, but 0.10% higher on half-MBB. These differences are smaller than the reported seed standard deviations; the three-seed experiment does not establish an objective advantage for either layer. Both QP methods pass all nine feasibility checks, whereas CNF-style passes one. These are initialization checks on fixed development geometries, not independent geometry generalization. The shared analytic cut and OC guidance are available to every method and are not FENCE-specific gains.

M Explicit Assembly Interface and Conformance

The shared frontend, `fence_contracts.py`, stores each linear functional as a sparse row block R, a name, and a coverage annotation. Basis callbacks provide evaluations or analytic derivatives; left multiplication composes selectors, finite differences, and quadrature. Declarations specify $\ell \le R\beta \le u$, with equal bounds denoting an equality. For prescribed coefficients $\beta_F = c$, assembly produces $\ell - R_F c \le R_{\bar{F}}\beta_{\bar{F}} \le u - R_F c$. The same object exports either two-sided sparse rows or canonical (A, b, G, h), and retains row provenance. It rejects nonfinite operators, invalid bounds and dimensions, and contradictions in constant rows created by substitution. General infeasibility remains the solver's responsibility.

Interface scope. Coverage annotations record the caller's argument; they are not machine-checked proofs. The frontend does not derive the heat gradient envelope, discover which topology supports intersect a forbidden region, or certify quadrature error; these are the application-specific constructions in Appendices I and K. It also does not linearize nonlinear physics or differentiate constraint assembly.

Post-experiment conformance. The reported experiments originally used task-specific assembly scripts. We subsequently implemented the shared frontend and checked it against those scripts using development inputs only. Those initial probes did not rerun training. The later complete heat replay in Appendix N exercises training and evaluation through the frontend without tuning or claiming new independent test performance. The audit checks nonzero fixed-coefficient substitution and bound signs, five invalid-input categories, and polynomial derivative and Gauss-quadrature probes. For the three sufficient topology geometries, exact integral rows agree and nine preseeded coefficient proposals produce projections within 2.23×10^{-15} of the specialized projection. For two development motions (walk1_subject1 and fight1_subject2), selector and difference matrices agree exactly and projected trajectories reproduce the saved pilot arrays. For two heat validation cases, the unnormalized 16,673-row systems agree within 3.56×10^{-15}; full-QP and constraint-exchange projections differ by at most 8.56×10^{-13}. Source and input hashes accompany `audit_compiler_contracts.py` and its results. These probes establish compatibility on the checked inputs, not universal correctness, runtime superiority, or additional empirical evidence for generalization.

N Coverage, Cost, and Frontend Execution

Frozen development protocol. The additional study fixes the first twelve tasks from validation generator seed 5200, the 16-mode sine basis, task objective, and Bernstein source and analytic power rows. Temperature covers use $(n_r, n_a) \in \{(8, 32), (16, 64), (32, 128), (64, 256)\}$. The sampled variant uses zero margin; the sufficient variant uses $\delta = r(1/(2n_r) + \pi/n_a)$. All configurations run through the shared compiler. Neither test cases nor method rankings are used to choose the cover. Clarabel uses feasibility and gap tolerances 10^{-13}. Exchange starts with the 33 source/power rows, adds at most eight violated temperature rows per iteration, and must scan every candidate before terminating at 10^{-8} physical residual. Its cap is 61 iterations. Full QP and exchange use identical objectives and rows.

Across all 96 configuration–task pairs, maximum full/exchange coefficient discrepancy is 3.9×10^{-10} and objective discrepancy is 4.0×10^{-14}. The sufficient variant passes all twelve dense/FD checks at every density. The sampled variant passes 8, 8, 10, and 10, respectively. These observations support the margin construction on this development set; finite exchange alone does not certify unsampled temperature values. The independent 192×192 FD check remains distinct from the continuum sine-field bound.

An additional equality through the same frontend. We append $\int (x - 0.5) q(x, y)\, dx\, dy = 0$, using exact sine moments and the finest sufficient temperature cover. This centers the source in x whenever its total power is nonzero; it is a linear equality even for the zero source. All twelve direct solves satisfy the combined contract, with maximum row residual 3.42×10^{-13} and twelve dense/FD passes. Mean objective increases from 5.50×10^{-3} to 8.34×10^{-3}. No model is retrained or selected for this diagnostic; it establishes executable composition on these inputs only.

Timing scope. Table 11 reports median solver times over the twelve tasks, including all row scans and every intermediate QP. Common compiler assembly and canonical-matrix extraction are outside these solver timings. A separate end-to-end comparison includes matrix extraction, encoder evaluation when applicable, and projection or task optimization; both methods use the finest sufficient contract and exchange. Three repetitions alternate method order on each task. Median

Table 11: **Temperature coverage at fixed source bounds.** Twelve development tasks. Paired entries are sampled / margin (S/M). Pass counts are out of 12. Times are median ms for the QP solve including exchange scans, excluding shared assembly.

Cover	Rows	$10^3 J$	Sampled pass	Margin pass	Exchange (ms)	Full QP (ms)
8×32	321	5.320/6.838	8/12	12/12	0.89/0.95	2.12/2.41
16×64	1,121	5.321/6.060	8/12	12/12	1.00/1.03	8.18/8.73
32×128	4,257	5.322/5.681	10/12	12/12	1.26/1.33	34.49/37.70
64×256	16,673	5.322/5.497	10/12	12/12	2.53/2.67	186.47/205.48

learned/direct times are 7.69/7.57 ms; mean times are 7.90/9.33 ms. Common assembly has median 16.41 ms. Timing differences at this scale are hardware-sensitive, and neither the medians nor the poorer learned objective establish amortization. Training costs are additional. This study does not support an inference-speed claim.

Full frontend training replay. The original heat experiments used task-specific assembly. To test actual execution through the shared frontend, we replay the complete FENCE heat training protocol for seeds 11–13: 128 training tasks, 200 steps, batch size eight, learning rate 10^{-3}, and the already frozen projection-distance weight 0.1. Every training projection is assembled by `ConstraintCompiler`; no saved coefficients substitute for training. All three models are then evaluated on the same previously used 96 tasks without tuning. The maximum matrix/scaling discrepancy from the original assembly is 1.14×10^{-13}; replayed test coefficients differ by at most 1.12×10^{-11} and retain all 288 dense/FD passes. This is end-to-end reproducibility evidence for heat, not a new independent test or a retroactive claim that the original experiments used the frontend. The original discrete retargeting and earlier topology experiments retain only the conformance evidence of Appendix M. The separate continuous-motion experiment in Appendix O uses it for every derivative construction and solve.

Sources, fixed protocol, raw fields, all configuration results, and complete replay histories are retained in `heat_coverage_value` and `heat_frontend_replay`.

O Continuous-Time Spline Derivative Coverage

This separate extension tests coverage in a temporal representation, beyond the sine-field heat construction. Development uses the four original development motions, `walk1_subject1`, `dance1_subject2`, `run1_subject2`, and `fight1_subject2`. The ten previously evaluated held-out motions are not used. The objective fits the same raw GMR angles at 150 times, with the same five raw anchors for 14 upper-body joints. The limits apply to true derivatives, $|q'(t)| \leq 2.5$ rad/s and $|q''(t)| \leq 20$ rad/s^2, for the full duration $149/30$ seconds. This differs from the finite-difference contract of Section 4.1.

Each joint has 150 open cubic B-spline coefficients. Internal knots are averages of three consecutive observation times, giving 147 spans, with twice-longer first and last spans. The common convex objective is $\frac{1}{2T}\|B\beta - q_{\mathrm{raw}}\|_2^2, T = 150$. All methods use the same basis, anchors, data, and objective. Each derivative construction is assembled by the shared frontend. These are direct optimization experiments, and the scope limitations of Section 3.4 apply.

Finite sufficient derivative bounds. For any subinterval $[a, b]$ of a knot span, q' is quadratic. Its three Bernstein coefficients, each linear in β, are

$$v_0 = q'(a), \qquad v_1 = q'(a) + \tfrac{b-a}{2} q''(a), \qquad v_2 = q'(b). \tag{16}$$

Since the Bernstein weights are nonnegative and sum to one, bounding all v_i by the velocity limit suffices throughout that interval. Acceleration is linear, so its two endpoint bounds are exact. Subdivision into two or four intervals tightens the velocity enclosure without changing the spline space. This is an application of the standard convex-hull property (Farouki, 2012; de Boor, 1978), not a new spline theorem.

We compare derivative sampling at the original 150 times and at 597 times, sufficient bounds with one or two subdivisions, and analytic constraint exchange. Exchange starts from knot-endpoint derivative constraints, locates all interior velocity extrema from the roots of q'', and adds violated first-derivative evaluation rows. It stops only after continuous separation finds no violation above its declared numerical tolerance, or reports a budget failure. This supplies a less conservative comparison than the fixed enclosure. A second solver also evaluates the identical one-span sufficient QP.

Coverage, quality, and cost. Table 12 reports all four motions. Sampled constraints are satisfied at their nodes, but speed can exceed the limit between them. The envelope covers these intervals with fewer rows than the 120-Hz sampler at its coarsest setting. Relative to analytic exchange, one- and two-subdivision envelopes keep mean objective overhead below 0.1%. Both samplers include every knot, so they cover linear acceleration exactly; their failures concern quadratic velocity.

Table 12: **Continuous derivative coverage on four development motions.** Pass requires raw-anchor error $\leq 10^{-8}$ rad and true derivative excess $\leq 10^{-5}$ in physical units. Rows are maximum two-sided interval rows per joint, including five anchor rows where applicable. Times are median seconds per 29-joint clip, including assembly and solves but excluding common basis construction and final extrema audit. They are implementation timings, not a solver-controlled speedup comparison.

Construction	Pass	RMS (°)	Max speed excess	Rows	Seconds
Sampled 30 Hz	0/4	1.80501	0.166667	305	0.125
Sampled 120 Hz	0/4	1.81012	0.008333	1199	0.354
FENCE (one-span envelope)	4/4	1.81116	1.8×10^{-11}	740	0.232
FENCE (two subdivisions)	4/4	1.81054	6.7×10^{-11}	1475	0.322
FENCE (four subdivisions)	4/4	1.81038	9.1×10^{-11}	2945	0.736
FENCE (analytic exchange)	4/4	1.81026	2.4×10^{-8}	375	0.313
Direct task QP (one-span)	4/4	1.81116	0	740	0.156

Acceleration excess is below 4×10^{-9} for every completed configuration. We verify extrema independently by converting scalar splines to piecewise power polynomials and evaluating endpoints and real critical points, cross-checked against 10,001 sampled times. Dense sampling alone is not the acceptance criterion. One-span FENCE matches Direct QP feasibility at slightly lower RMS and median time (0.220 s vs. 0.280 s). These results quantify the coverage–quality tradeoff across enclosure strategies.

Numerical controls. Averaged knots give collocation condition number 3.89. Fixed systems use OSQP at 10^{-10}; exchange uses Clarabel at 10^{-10} with the whole objective scaled by 150. Each solve has a four-second limit; exchange permits 60 rounds and 30 seconds per joint. Constraints and acceptance tolerances are unchanged. Earlier exchange configurations with default solver settings failed on 11 joints with OSQP and two with tighter Clarabel; these failures guided the final solver selection and tolerance tuning. Independent sufficient-QP objectives agree within 5.3×10^{-13}, with coefficient discrepancy 2.34×10^{-5}, slightly above the synthetic preflight target 10^{-5}. Failed configurations and all solver histories are preserved.

Frozen validation on seven new clips. After development, we freeze the basis, all solver settings, and the same physical limits before reading seven additional BVH files. For each previously unused action category (aiming, fall-and-get-up, ground, jumps, obstacles, push-and-stumble, sprint), we select the lexicographically first available file among subjects 3–5: `aiming2_subject3`, `fallAndGetUp2_subject3`, `ground2_subject3`, `jumps1_subject5`, `obstacles3_subject3`, `pushAndStumble1_subject3`, and `sprint1_subject4`. Unmodified GMR runs from frame 0; frames 60–209 provide the raw inputs. None of these files is in the original ten-clip discrete evaluation. All seven are retained, with no post-validation tuning or replacement. This tests new clips and action categories for the same G1 embodiment, not new-subject or physical-execution generalization.

All 49 method–clip configurations complete within the frozen settings; total runtime, including GMR generation, is 50.8 seconds. The one-, two-, and four-subdivision envelopes have mean per-clip objective overheads of 0.0727%, 0.0178%, and 0.00921% relative to exchange. Their worst-case overheads are 0.2345%, 0.0497%, and 0.0217%. An additional saved-array audit reconstructs the velocity quadratic from its endpoint and midpoint values, separately locating its extrema without the original power-polynomial routine; all 49 feasibility decisions agree. The independent QP and exchange also pass 7/7, confirming that the coverage guarantee is a property of the constraint construction rather than a specific solver.

Independent evaluation package. A NumPy-only de Boor and derivative evaluator (de Boor, 1978), independent of SciPy, GMR, and training code, reproduces all 49 saved-array decisions in Python 3.11 with NumPy 1.26.4. Maximum speed, acceleration, and objective discrepancies are 8.9×10^{-16}, 3.6×10^{-15}, and 4.4×10^{-19}. The relocatable package verifies input hashes and rejects corrupted copies; it checks saved-array evaluation, not retraining or a new validation set.