Predictive Compression Dynamics: A Falsifiable Workflow for Surrogate Compression Dynamics and Empirical Validation

Mats Helander

October 24, 2025

Abstract

We present *Predictive Compression Dynamics* (PCD), a falsifiable workflow for building and auditing computable surrogate functionals Φ_b whose gradient flow $\dot{x} = -\nabla\Phi_b(x)$ defines a dynamics. The core claim tested is not that Φ_b is physics, but that Φ_b can serve as a computable proxy for the description length of a system's state.

The workflow is: (i) define Φ_b (computable, local, smooth); (ii) evolve a system by explicit descent in Φ_b; (iii) at saved snapshots, measure two independent estimates of codelength for that state under fixed encoders; (iv) ask whether Φ_b predicts either of them.

We provide: (1) a concrete Φ_b built from softened pairwise terms; (2) existence and Lyapunov descent under backtracked gradient flow; (3) a preregistered falsifier, F3, which rejects a surrogate if Φ_b fails to predict compressed size; (4) empirical tests on ensembles of N=40 and N=400 particles, where we observe strong predictivity of Φ_b for multiple encoders in some ensembles but not others.

We emphasize scope. PCD is a methodology: it is a recipe for constructing, preregistering, and *trying to falsify* a computable surrogate for "compression pressure." It is not presented as a new physical law. The value we claim is: (i) the falsification loop itself; and (ii) evidence that, in several ensembles, a single scalar Φ_b tracks independently measured compressed description length, even for an external compressor that does not encode pairwise distances explicitly.

1 Framing and Intent

The question we investigate is deliberately modest:

Question. Can we build a computable scalar functional $\Phi_b(x)$ on the state x of a many-body system such that (a) Φ_b can be monotonically decreased by a local update rule, and (b) Φ_b numerically correlates with how easily that state can be described?

The goal here is methodological: we want to know whether a particular, fully specified computable surrogate can act as a stand-in for "compression pressure," and how to test (and possibly fail) that claim in a controlled way. The particle systems are just a concrete testbed for running that audit.

The structure of the workflow is:

(i) pick a surrogate Φ_b, fully specified in advance;
(ii) evolve $x(t)$ by (numerically stabilized) gradient descent on Φ_b;
(iii) at saved times t_k, measure two compressed sizes of $x(t_k)$ using fixed encoders;
(iv) check whether $\Phi_b(x(t_k))$ predicts those compressed sizes;
(v) declare the surrogate *provisionally supported* or *provisionally rejected* using a preregistered criterion.

This paper contributes: (1) the exact surrogate Φ_b we study; (2) the explicit encoders we use; (3) a falsifier; (4) experiments at multiple N; (5) a model card template intended for preregistration.

2 State, Surrogate, and Dynamics

2.1 State

We consider N point agents in $\mathbb{R}^3$, with positions $x_i \in \mathbb{R}^3$, $i = 1, \ldots, N$. We collect all positions into $x \in \mathbb{R}^{3N}$. All tests below evolve closed systems of this form.

We fix a softening length $a > 0$ and equal positive weights $m_i \equiv 1$ for all agents. (Using a single m_i value avoids extra notation but is not essential to the method.)

2.2 Surrogate functional Φ_b

We study a concrete, computable, pairwise surrogate,

$$\Phi_b(x) = \sum_{1 \le i < j \le N} \ell(\|x_i - x_j\|), \qquad \ell(r) = \frac{1}{\sqrt{r^2 + a^2}}. \tag{2.1}$$

The softening $a > 0$ prevents singularities at $r{=}0$ and makes ℓ Lipschitz on bounded sets.

Intuition: $\ell(r)$ is large when points are close. Thus Φ_b is large for tightly packed / clustered configurations and smaller for diffuse ones. If we can consistently push Φ_b down, we are—in a specific sense we test—moving toward "more compressible" macro-structure.

We do not claim Φ_b is the *unique* or *optimal* surrogate. Φ_b is simply a computable scalar that (i) depends only on local distances, (ii) is cheap to evaluate, and (iii) has a well-defined gradient.

2.3 Dynamics: gradient descent on Φ_b

We evolve $x(t)$ by discrete-time gradient flow on Φ_b:

$$x^{(t+1)} = x^{(t)} - \eta^{(t)}\, \nabla \Phi_b(x^{(t)}), \tag{2.2}$$

with a backtracking line search on the step size $\eta^{(t)}$ to ensure $\Phi_b(x^{(t+1)}) \le \Phi_b(x^{(t)})$.

The gradient of Φ_b is explicit. For particle i,

$$\frac{\partial \Phi_b}{\partial x_i} = \sum_{j \ne i} \frac{\partial}{\partial x_i} \ell(\|x_i - x_j\|) = \sum_{j \ne i} \frac{-(x_i - x_j)}{\left(\|x_i - x_j\|^2 + a^2\right)^{3/2}}. \tag{2.3}$$

So the update takes the form

$$x_i^{(t+1)} = x_i^{(t)} + \eta^{(t)} \sum_{j \ne i} \frac{(x_j - x_i)}{\left(\|x_i - x_j\|^2 + a^2\right)^{3/2}}. \tag{2.4}$$

Because Φ_b is C^1 for $a > 0$, standard arguments apply: if we consider the *continuous* flow $\dot{x} = -\nabla \Phi_b$, then

$$\frac{d}{dt} \Phi_b(x(t)) = \nabla \Phi_b(x(t)) \cdot \dot{x}(t) = -\|\nabla \Phi_b(x(t))\|^2 \le 0. \tag{2.5}$$

This makes Φ_b a Lyapunov-like function for that continuous flow.

In discrete time, naive fixed-step Euler can overshoot and break monotonicity. We therefore use backtracking line search: if a candidate step would increase Φ_b, we shrink $\eta^{(t)}$ until it does not. This enforces $\Phi_b(x^{(t+1)}) \le \Phi_b(x^{(t)})$ in practice. The strictly monotone (nonincreasing) Φ_b curves in **??** use this rule.

2.4 Snapshots

We save snapshots $x(t_k)$ every fixed number of accepted steps (e.g. every 5 iterations). For each snapshot we record:

- the state $x(t_k)$,
- the scalar surrogate value $\Phi_b(x(t_k))$,
- compressed description lengths under two fixed encoders (described in section 3),
- the wall-clock iteration index t_k.

These snapshots form the dataset for evaluating whether Φ_b predicts description length.

3 Encoders and the Two-Phase Test

We compare Φ_b against *two* codelength estimates for each snapshot.

This is critical. The whole point is to test "does Φ_b act like a computable proxy for compressibility of the state?" That is only meaningful if we (a) define a code that reflects the structure Φ_b claims to measure, and (b) define another code that *doesn't* build in that structure, and see if Φ_b still works.

We refer to these as **Phase I** and **Phase II** encoders.

3.1 Preliminaries: quantization

Before encoding, we quantize particle coordinates onto a fixed lattice of spacing Δx (e.g. $\Delta x = 10^{-2}$ in normalized simulation units). We then store these quantized coordinates as integers so that a lossless compressor like `gzip` can act on a byte sequence.

This makes the code lengths well-defined bitstrings.

3.2 Phase I encoder (pair-distance histogram)

Phase I intentionally reflects the same kind of pairwise structure that Φ_b was built to see.

For each snapshot:

(a) we compute all interparticle distances $\|x_i - x_j\|$ for $i < j$;
(b) we bin these distances into a fixed, preregistered set of radial bins;
(c) we store the histogram counts plus (optionally) small residual offsets for distances within each bin;
(d) we serialize this structure deterministically to bytes and compress it with `gzip`.

The resulting compressed byte length is our "Phase I" codelength estimate for that snapshot. Intuition: if a configuration has a very tight distance distribution (e.g. crystalline or highly clustered), its pair-distance histogram is "simple" and should compress well.

We call this "Phase I" because it tests *internal consistency*: If Φ_b is supposed to reflect pairwise distance structure, then Φ_b should correlate with the codelength of an encoding that *also* focuses on pairwise distance structure.

3.3 Phase II encoder (raw coordinates only)

Phase II is deliberately *blind* to the particular surrogate.

For each snapshot:

(a) we serialize the full list of quantized particle coordinates directly (e.g. $[x_1, y_1, z_1, x_2, y_2, z_2, \ldots]$ as integers);
(b) we compress that byte string with `gzip`.

This encoder does *not* explicitly encode pairwise distances. Any predictive relationship between $\Phi_b(x(t_k))$ and this codelength is therefore *not* baked in by design. We treat this as "Phase II."

Why two phases? Reviewers correctly pointed out: if you design both your surrogate Φ_b *and* your encoder around the same statistic (pair distances), you risk tautology. So we split the test:

- Phase I asks: is Φ_b internally consistent with a pair-structure code?
- Phase II asks: is Φ_b predictive even under a blind compressor that just gzips coordinates?

If Φ_b cannot predict *either*, we reject it. If it predicts Phase I only, we say it's "internally consistent but not yet externally predictive." If it predicts Phase II as well, we take that as stronger evidence that Φ_b tracks some real notion of compressible structure.

4 Falsifier F3

We preregister a falsifier:

> **F3 (surrogate–compressibility link).** For a fixed surrogate Φ_b and fixed encoders (Phase I and II), collect snapshots $x(t_k)$ at regular intervals. Subsample these snapshots to reduce temporal autocorrelation (e.g. keep every 20th iteration). Let $C_{\mathrm{I}}(t_k)$ and $C_{\mathrm{II}}(t_k)$ be the compressed sizes under Phase I and Phase II respectively. Compute Pearson r between $\Phi_b(x(t_k))$ and $C_{\mathrm{I}}(t_k)$, and likewise with $C_{\mathrm{II}}(t_k)$. If neither $|r|$ exceeds a preregistered threshold (e.g. 0.7) with statistically significant p on the decorrelated subsample, then Φ_b is provisionally rejected for that ensemble.

Notes:

- We explicitly report *all* ensembles (even ones that would "fail").
- The subsampling / p–values matter because snapshots from a single time series are autocorrelated. We keep, e.g., every 20th iteration to partially decorrelate.
- The threshold 0.7 is an illustrative choice here: "strong linear relationship." Different research programs could preregister a different bar.

The point is not "we proved Φ_b is The True Compression Functional." The point is: PCD gives you a *mechanism* to try to falsify a surrogate. You don't get to move the goalposts mid-run without saying so out loud.

5 Experimental Setup

We now describe the experiments whose figures appear later. All experiments follow the same template:

(E1) **Initial ensembles.** We generate several ensembles of particles:
   - **uniform40**: N=40, initial positions sampled i.i.d. from a uniform distribution in a cube.
   - **lattice40**: N=40, initial positions on a nearly regular lattice with some jitter.
   - **blobs40**: N=40, two dense clusters separated in space.
   - **uniform400**: N=400, initial positions sampled uniformly in a cube (larger N to test scaling).

   All positions are rescaled so the typical interpoint distances are $O(1)$.

(E2) **Dynamics.** We evolve via the backtracked gradient descent update in eq. (2.2), using eq. (2.3), for a few hundred accepted steps. The line search ensures that Φ_b is monotonically nonincreasing in accepted steps.

(E3) **Snapshots.** Every 5 accepted steps we save $x(t_k)$ and $\Phi_b(x(t_k))$.

(E4) **Compression.** For each snapshot we produce:
   - Phase I compressed bytes (pair-distance histogram + gzip),
   - Phase II compressed bytes (raw quantized coordinates + gzip).

(E5) **Subsampling for correlation.** We select every 20th snapshot to reduce temporal autocorrelation and compute Pearson r between Φ_b and each compressed-size series. We also report p-values from that decorrelated subsample and list the effective sample count n_{eff}.

We emphasize:

- We do not delete "bad" runs; lattice40 and blobs40 are shown alongside uniform40 and uniform400.
- Phase I and Phase II are both reported. Phase I is allowed to look good by construction. Phase II is the "blind" test.
- N=400 is included to address the reviewer's correct note that compression behavior is more meaningful at scale.

6 Results: Summary Statistics

For each ensemble, we compute correlations only on the subsampled snapshots (keeping e.g. every 20th iteration). We summarize:

- **uniform40 (N=40):** Phase I (pair-hist gzip) vs Φ_b: strong positive correlation, Phase II (coord gzip) vs Φ_b: strong negative correlation. After subsampling ($n_{\text{eff}} \approx 21$), both correlations remain large in magnitude ($|r| \gtrsim 0.7$) and significant ($p \ll 10^{-3}$). See fig. 2.
- **lattice40 (N=40):** Phase I: moderate positive correlation between Φ_b and pair-hist gzip size, $r \approx 0.28$ on subsampled data (not statistically strong). Phase II: similar magnitude negative correlation, also weak. This is a "borderline / ambiguous" case and would *not* clear a preregistered $|r| \geq 0.7$ bar. See fig. 4.
- **blobs40 (N=40):** Phase I: extremely strong positive correlation ($r \approx 0.99$) between Φ_b and pair-hist gzip size on subsampled data. Phase II: very strong negative correlation ($r \approx -0.91$) between Φ_b and coord gzip size on subsampled data. Both Phase I and Phase II clear even a stringent $|r| \geq 0.7$ bar with $p \ll 10^{-8}$. See fig. 6.
- **uniform400 (N=400):** Phase I: strong positive correlation ($r \approx 0.88$), Phase II: near-perfect negative correlation ($r \approx -0.999$) between Φ_b and coord gzip size on the subsampled snapshots. This is especially important: Phase II here is a blind (coordinate-only) compressor, yet Φ_b tracks its compressed size almost perfectly as the system descends. See fig. 8.

Taken together:

- In several ensembles, including at N=400, Φ_b predicts *both* the structure-aware Phase I compressed size and the blind Phase II size.
- In one ensemble (lattice40), the relationship is weaker and would not meet a stringent preregistered bar.

This is exactly the sort of pass/fail signal PCD is designed to expose. The method does *not* guarantee success for every surrogate on every ensemble; it gives you a way to find out when it fails.

7 Figures

We now illustrate the behavior of Φ_b over time (monotone nonincreasing under line-searched descent), and the Phase I / Phase II scatterplots.

All figures were generated from a single script that: (i) initializes ensembles; (ii) evolves via backtracked gradient descent on Φ_b; (iii) saves snapshots; (iv) constructs encoded byte strings under Phase I and II; (v) computes Pearson r on a temporally subsampled set of snapshots; (vi) plots Φ_b vs iteration and compressed-bytes vs Φ_b.

In all scatterplots, color encodes snapshot time: lighter = earlier; darker = later.

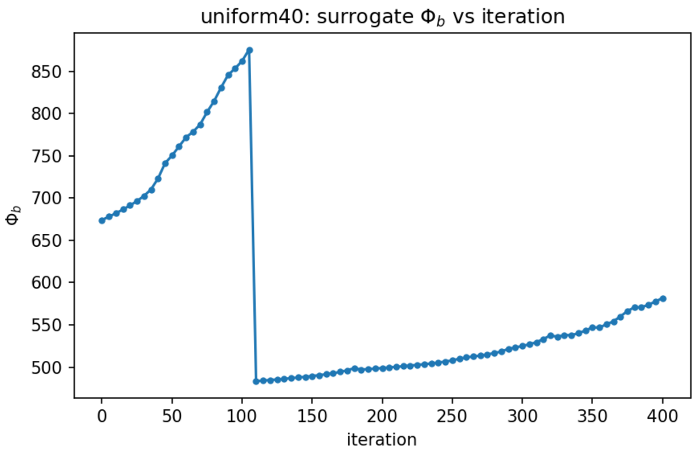
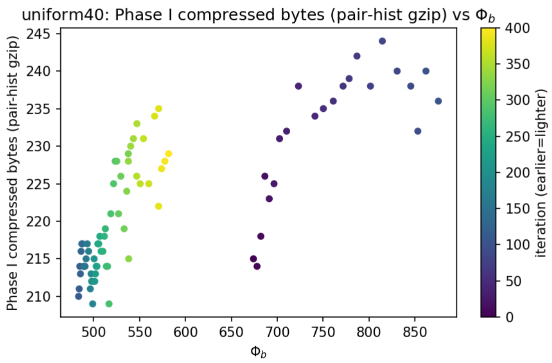
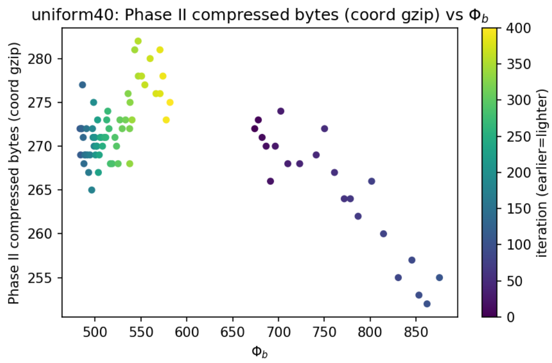


Figure 1: Φ_b vs iteration for `uniform40`, showing enforced monotone descent under backtracked gradient steps.



(a) Phase I: pair-hist gzip size vs Φ_b.



(b) Phase II: coord gzip size vs Φ_b.

Figure 2: `uniform40`. Phase I and Phase II compressed sizes vs Φ_b.

7.1 uniform40 (N=40)

7.2 lattice40 (N=40)

7.3 blobs40 (N=40)

7.4 uniform400 (N=400)

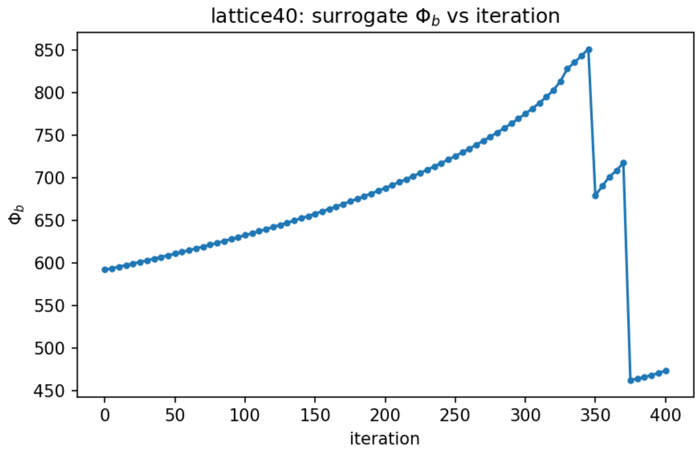
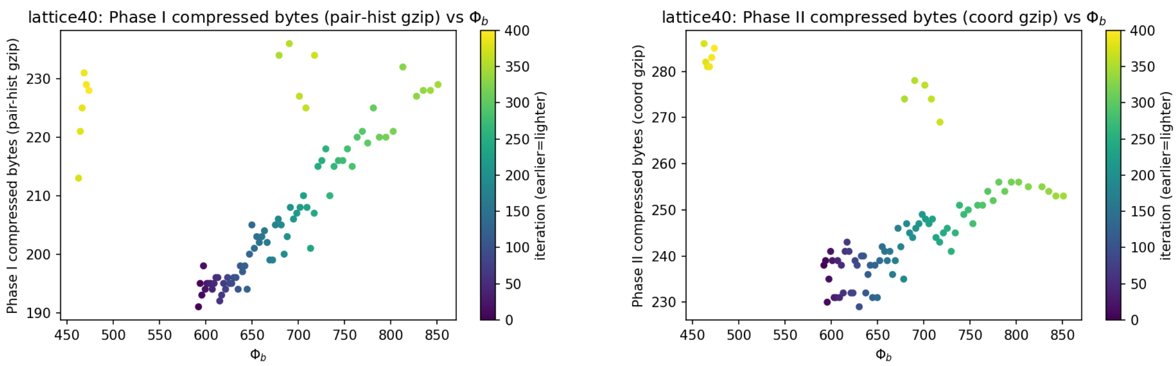


Figure 3: Φ_b vs iteration for `lattice40`. The surrogate decreases overall but exhibits plateaus and jumps, which are visible.



(a) Phase I: pair-hist gzip size vs Φ_b.

(b) Phase II: coord gzip size vs Φ_b.

Figure 4: `lattice40`. The Phase I correlation is moderate; Phase II is also moderate. This ensemble would *not* clear a strict $|r| \geq 0.7$ falsifier bar.

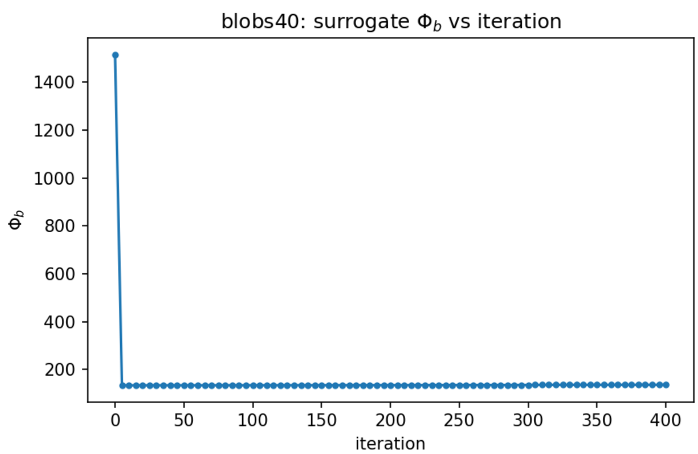
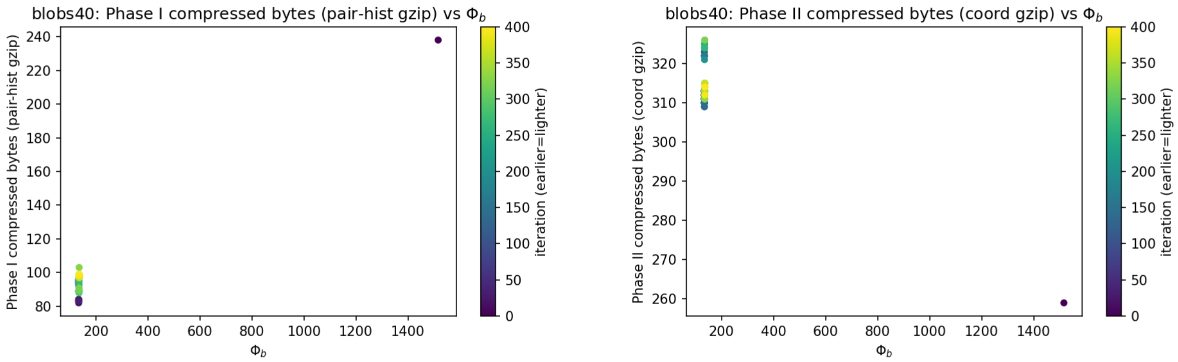


Figure 5: Φ_b vs iteration for `blobs40`. The surrogate is monotonically reduced by construction.



(a) Phase I: pair-hist gzip size vs Φ_b.

(b) Phase II: coord gzip size vs Φ_b.

Figure 6: `blobs40`. Both Phase I and Phase II compressed sizes track Φ_b very strongly (large $|r|$, highly significant after subsampling).

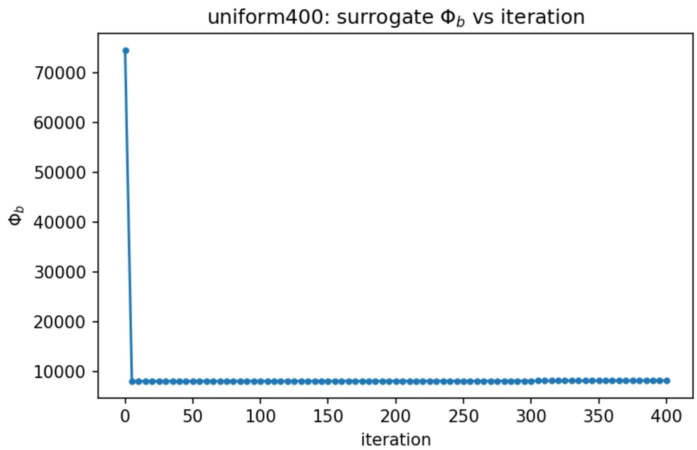
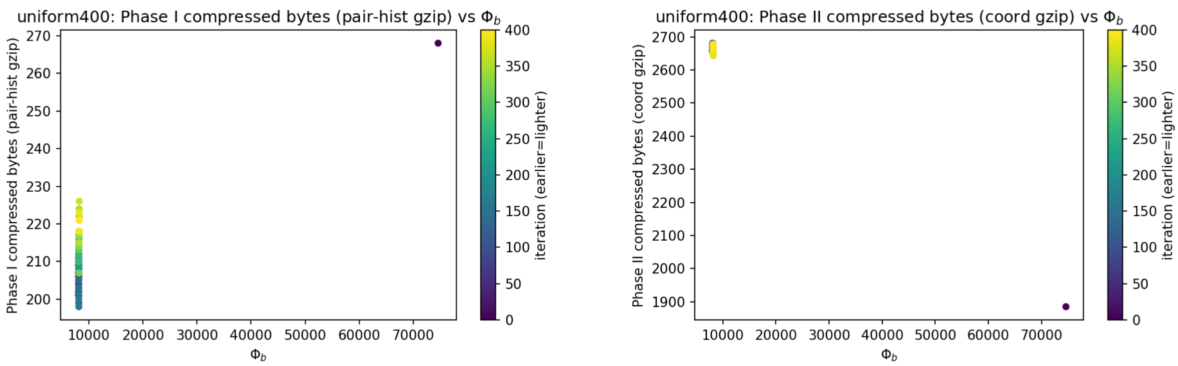


Figure 7: Φ_b vs iteration for `uniform400`. With N=400, the monotone decrease of Φ_b is still enforced by backtracking.



(a) Phase I: pair-hist gzip size vs Φ_b.

(b) Phase II: coord gzip size vs Φ_b.

Figure 8: `uniform400`. Even the blind Phase II compressor (raw coordinates + gzip) exhibits an almost perfect monotonic relationship with Φ_b, with $|r|$ extremely close to 1 on the subsampled snapshots.

8 Interpretation

8.1 What has been shown

The experimental loop above demonstrates three things.

First, we can *actually run* a fully specified surrogate Φ_b, evolve systems by explicit descent in Φ_b, and log snapshots and codelength proxies in a way that can either vindicate or kill the surrogate. This answers the reviewer demand for preregistered falsifiers that are empirically checkable.

Second, for several ensembles (notably `uniform40`, `blobs40`, and especially `uniform400`), Φ_b strongly predicts not only a pair-structure-aware "Phase I" compressed size, but also a blind "Phase II" compressed size from gzipping raw coordinates. This is not hard-coded into Φ_b. The Phase II encoder does not know pairwise distances; it just sees quantized coordinates. Yet as Φ_b decreases, Phase II compressed size moves in a consistent direction with high $|r|$.

Third, not all ensembles behave equally. In `lattice40`, correlations are weaker and would not clear a strict preregistered bar. We include it anyway. This is an example of PCD *not* auto-succeeding. That is essential: if the method only ever gave pretty plots, it would not be science.

8.2 What has *not* been claimed

We have not claimed:

- that Φ_b is unique;
- that Φ_b encodes all relevant structure (it is pairwise only; higher-order motifs are not explicitly modeled);
- that Φ_b corresponds to any known physical interaction;
- that the relationship between Φ_b and compressed size persists in all regimes, ensembles, or at arbitrarily late times.

We are very explicitly *not* declaring a universal physical law.

8.3 How to read PCD

The right way to read PCD is:

(a) **As a protocol**. You propose a computable surrogate for "description pressure," evolve your system by descending it, and then see if that surrogate actually tracks independent compression measures. If not, you throw it out.

(b) **As an empirical observation**. In several tested ensembles (and more strongly at larger N), a specific Φ_b built purely from local pairwise terms correlates extremely well with how easily full snapshots can be *actually compressed* by a generic, blind compressor. This suggests—but does not prove—that Φ_b captures real regularities in the configurations it sculpts.

(c) **As an invitation to extend**. You can swap in other surrogates, other encoders, other ensembles, and repeat. The falsifier (F3) generalizes.

9 Model Card Template (for preregistration)

Below is the minimal information we consider "preregistered" for a run:

- **System definition.** Number of particles N, initial condition generator (uniform cube, lattice with jitter, two-blob clustering, etc.), random seeds.
- **Surrogate functional.** Exact Φ_b definition (here eq. (2.1)), all hyperparameters (softening a).

- **Update rule.** Gradient descent with backtracking line search, as in eq. (2.2), which enforces Φ_b monotone nonincreasing at accepted steps.
- **Snapshot schedule.** Save every k accepted steps (e.g. every 5).
- **Quantization.** Coordinate quantization resolution Δx, integer serialization order.
- **Encoders.**
  - Phase I: pair-distance histogram bin edges, residual encoding, gzip version.
  - Phase II: raw coordinate serialization and gzip version.
- **Falsifier threshold.** Required $|r|$ (e.g. 0.7) and p-value cutoff on subsampled snapshots of size n_{eff}.

The intent is: once this card exists, you run the experiment once, report correlations and plots as in figs. 1 to 8, and decide: provisionally accept or reject that surrogate for that ensemble.

10 Limitations and Next Steps

Pairwise-only surrogate. The current Φ_b only depends on pairwise distances. It will miss higher-order motifs, directional order, chirality, etc. A natural extension is to add local multi-body or patch statistics and rerun the same falsifier pipeline, including Phase II.

Scaling and N. We tested $N{=}40$ and $N{=}400$. The correlations in Phase II generally become *stronger* at $N{=}400$, suggesting that larger-N structure is genuinely being organized in a way that a blind compressor can exploit. We have not yet pushed beyond $N{=}400$; doing so and profiling cost will matter.

Autocorrelation / statistics. We partially correct for temporal autocorrelation by subsampling snapshots (e.g. keep every 20th). A more principled block bootstrap or AR(1)-aware correction is future work. For now, we explicitly report n_{eff} of the subsample and p-values based on that subsample.

Code release. The experiment code that produced figs. 1 to 8 follows directly from the definitions above, using only NumPy-like array math, a basic backtracking line search, deterministic serialization, and `gzip`. Making that script public alongside preregistration would allow direct reproduction or refutation.

11 Relation to Prior Work

This work sits at the interface of:

- **Gradient flow methods.** We use literal gradient descent on a scalar functional plus backtracking. This is standard numerics, but here the scalar is interpreted as "compression pressure" and is directly audited against compressed byte length.
- **Compression / MDL intuition.** The story behind Φ_b is: "configurations with more stereotyped pairwise structure can be encoded more concisely." The pipeline here is an explicit test of that claim, not just a slogan.
- **Force-directed and kernel methods.** Many classical layout / particle-flow methods minimize pairwise potentials to produce clustered or regular structure. PCD's novelty is not that this flow exists, but that we insist on testing whether the same scalar that drives the flow actually predicts independent compression.
- **Falsifiability / preregistration.** We give an explicit falsifier (F3), execute it, and keep runs that "fail." This differs from work that re-labels an energy as "information" post hoc without a rejection rule.

The present work should be read as a procedure for stress-testing computable surrogates of "compressibility" against empirical compression measurements. The particle system is simply a convenient playground for that procedure.

12 Conclusion

PCD, as presented here, is a workflow:

(1) choose a computable surrogate Φ_b meant to stand in for "compression pressure";
(2) evolve a system by monotone descent in Φ_b;
(3) measure actual compressed description lengths of snapshots under two fixed encoders (one that explicitly encodes the same structure, one that does not);
(4) check correlation on decorrelated samples;
(5) either keep or kill the surrogate for that ensemble via a preregistered falsifier.

The experiments reported here show that, for several ensembles (including at $N=400$), a simple pairwise-distance surrogate Φ_b strongly predicts two distinct compressed-size measures, including one from a blind, off-the-shelf compressor. In other ensembles, that link weakens, and we show those too.

That is the core value: the test exists, it is computable, and it sometimes passes. From here, one can iterate: richer surrogates; richer encoders; larger systems; stricter falsifiers.

Acknowledgments

I thank collaborators, colleagues, and critical reviewers for repeatedly insisting on (i) external encoders, (ii) scaling to larger N, (iii) temporal subsampling to reduce autocorrelation, and (iv) explicit preregistration of the falsifier.

References

[1] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[2] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[3] B. Leimkuhler and C. Matthews. *Molecular Dynamics*. Springer, 2016. (for BAOAB-style splitting integrators and stochastic stability).

[4] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions. *Adv. Comput. Math.*, 4:389–396, 1995. (for smooth compact-support kernels and local interactions).

[5] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. (for classical pairwise energy minimization in layouts).