DUPLICATED-GRAPH TRAINING FOR MACHINE LEARNING FORCE FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning force fields now approach quantum-mechanical accuracy at a small fraction of its cost, yet every training label still requires an electronic-structure calculation, so how much a model learns from each label determines how far it reaches. Nearly all such models share one interface to their data: a labeled conformation is converted into a single cutoff graph, and that graph is everything the network ever sees of it. We revisit that interface with duplicated-graph training, which pairs each labeled conformation with a rigidly translated copy of itself and applies one cutoff rule to both, so that atoms interact within each copy and across the two. Rigidity leaves every intramolecular distance and angle intact, so the reference energy and forces remain exact labels, while the geometry between the copies varies with the translation. Resampling the translation therefore lets a single label constrain the model on a family of interaction geometries, with no additional electronic-structure calculations and no change to the architecture. Applied to MACE, duplicated graphs lower force error on every system we study: by 16.4% on average across the ten rMD17 molecules and by 9 to 20% on the seven MD22 systems. The gains are largest where the test conformations depart from the training distribution, growing on 3BPA from 4.57% at 300 K to 16.55% at 1,200 K. Distillation transfers the improvement to a model that runs on an ordinary graph at unchanged inference cost, retaining a 15.6% average force reduction on rMD17.

1 INTRODUCTION

Molecular dynamics (MD) simulations of proteins, catalysts and materials rely on a force field to predict energies and forces from atomic positions. Classical force fields are fast but cannot describe bond breaking or subtle electronic effects, while quantum-mechanical methods such as density functional theory are accurate at a cost that grows steeply with system size. Machine learning force fields (MLFFs) bridge this gap by learning the potential energy surface from quantum-mechanical reference calculations (Behler & Parrinello, 2007; Unke et al., 2021), reaching close to that accuracy at a small fraction of the evaluation cost (Batzner et al., 2022; Batatia et al., 2022). However, obtaining each energy–force label still requires an expensive electronic-structure calculation, limiting the available training data. This motivates training methods that make more effective use of each labeled configuration without requiring additional reference calculations.

Nearly all modern MLFFs share a common interface to the data. A configuration is converted into a graph whose nodes are atoms and whose edges connect atom pairs within a cutoff radius, and a message-passing network maps this graph to an energy whose negative gradient gives the forces. Most progress in data efficiency has come from the two ends of this pipeline. On the model side, equivariant architectures build geometric symmetries into the message-passing operators (Batzner et al., 2022; Batatia et al., 2022). On the data side, active learning chooses which configurations to label (Zhang et al., 2020), and pretrained foundation potentials amortize labels across chemical space (Kovács et al., 2025; Batatia et al., 2025). The interface between them has received far less attention: each labeled configuration still enters training through exactly one graph, the one produced by the cutoff rule applied to its coordinates. This raises a simple question. *Can a single labeled configuration supervise the model through more than one interaction geometry?*

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.

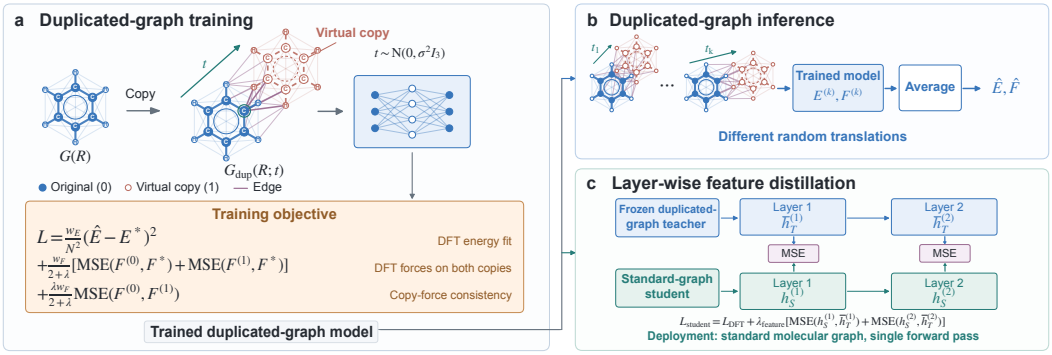

Figure 1: Overview of duplicated-graph training and deployment. (a) A labeled molecular graph and its rigidly translated duplicate form one interaction graph; the unchanged DFT energy and forces supervise both copies, with a copy-force consistency term. The network architecture is unchanged. (b) At inference, averaging predictions over different random translations uses the trained duplicated-graph model. (c) Alternatively, a standard-graph student learns the teacher's corresponding hidden features layer by layer while retaining DFT supervision, enabling a single-forward-pass deployment on the standard molecular graph.

The obvious way to vary what the network sees is to perturb the atomic coordinates, and this is the route taken by denoising objectives and by Taylor-expansion methods (Godwin et al., 2022; Zaidi et al., 2023; Liao et al., 2024; Cooper et al., 2020; Takamoto et al., 2025). Perturbing coordinates, however, changes the physical configuration, so the reference energy and forces no longer apply exactly. Each such method must therefore construct a new target, whether a noise-reconstruction objective or an approximate label from local derivatives, and the quality of that target limits the benefit. An ideal augmentation for force-field training would vary the interaction geometry presented to the network while leaving the labels exactly valid.

We introduce *duplicated-graph training*, which achieves this with a rigidly translated duplicate of the molecule. At every training step, we copy the labeled configuration, displace the copy by a random translation t, and build a single cutoff graph over the original and duplicated atoms together (Figure 1). Within each copy, all distances and angles are unchanged. Between the two copies, a cutoff search adds cross-copy edges whose relative displacements $(r_j - r_i) + t$ depend on the sampled offset. The network therefore processes a denser graph in which each atom interacts both with its usual neighbors and with a shifted image of the whole molecule, and the offset is resampled at every visit. Because the internal geometry of each copy is exact, both copies are supervised with the original energy and force labels, and no additional reference calculations are needed. The two copies share all network weights and exchange messages throughout message passing. Training fits the copy-averaged force to the reference and penalizes disagreement between the two copies. For ordinary-graph deployment, we distill intermediate representations from the duplicated-graph teacher into a student that uses the standard molecular graph.

We evaluate duplicated-graph training with MACE (Batatia et al., 2022) on rMD17, MD22, and 3BPA (Christensen & von Lilienfeld, 2020; Chmiela et al., 2023). At a budget of 950 training configurations per molecule, the duplicated graph reduces force MAE on all ten rMD17 tasks and all seven MD22 systems in our evaluation. On 3BPA, the relative force-RMSE reduction grows from 4.57% at 300 K to 16.55% at 1,200 K. Distillation transfers a force-accuracy gain to ordinary-graph students on all ten rMD17 tasks.

In summary, our contributions are as follows:

- We propose duplicated-graph training, which couples a molecule with a rigidly translated duplicate in a single cutoff graph. It varies the interaction geometry seen by the network while every energy and force label remains exact, and it requires no change to the backbone architecture and no additional reference calculations.

- We analyze the resulting training objective. We show that the copy-averaged force is the exact gradient of the averaged joint energy, and that the copy-supervised force loss decomposes into a fit to the reference plus a penalty on copy disagreement, which corresponds to the energy's response to relative duplicate motion.
- We show force-accuracy improvements for MACE across rMD17, MD22, and temperature-shifted 3BPA, and compare the full duplicate with coordinate perturbation and sparse auxiliary nodes.
- We distill the duplicated-graph teacher into a student on the standard molecular graph and observe lower force error than the MACE baseline on all ten rMD17 tasks.

2 Related Work

Data-efficient molecular force fields. Data efficiency has been pursued from two directions: architectures that build physical symmetries in, as in NequIP and MACE (Batzner et al., 2022; Batatia et al., 2022), and strategies that reduce the labeling burden by choosing which configurations to label (Zhang et al., 2020) or by amortizing labels across chemical space in transferable and foundation potentials (Kovács et al., 2025; Batatia et al., 2025). Duplicated-graph training holds the labels, the architecture and the initialization fixed and changes only the graph through which each configuration enters training, so it composes with all of these rather than competing with them.

Geometry perturbation and consistency objectives. A second direction extracts more signal from fixed data by perturbing coordinates. Denoising is a standard pretraining objective for 3D molecular models (Zaidi et al., 2023), with refinements to the noise distribution (Feng et al., 2023; Ni et al., 2024); Noisy Nodes applies it during training and DeNS extends it to non-equilibrium structures (Godwin et al., 2022; Liao et al., 2024), while Taylor-expansion and PIWSL objectives exploit local derivatives instead (Cooper et al., 2020; Takamoto et al., 2025), and consistency regularization penalizes disagreement between stochastic passes over one input (Liang et al., 2021). Perturbing coordinates changes the physical configuration, so each of these methods must define a new target. A rigid duplicate preserves every intramolecular distance and angle, leaving the reference energy and forces exact for both copies, and the resulting disagreement penalty acts between copies that exchange messages inside one graph rather than between independent forward passes.

Augmenting the message-passing graph. A third direction modifies the graph itself: dropping edges regularizes deep GNNs (Rong et al., 2020), rewiring relieves over-squashing (Alon & Yahav, 2021; Topping et al., 2022), Half-Hop inserts intermediate nodes, and NodeDup duplicates nodes and whole graphs (Azabou et al., 2023; Guo et al., 2025). In atomistic modeling, auxiliary representations extend local message passing through Fourier-space channels, molecular fragments, mesh points and global encodings (Kosmala et al., 2023; Li et al., 2024; Wang et al., 2024; Caruso et al., 2026). Multi-view methods either encode views separately and compare them contrastively (You et al., 2020; Wang et al., 2022) or couple two graphs by cross-graph attention (Li et al., 2019). The auxiliary nodes of a duplicated graph are geometric: they occupy the same three-dimensional space as the atoms, so cross-copy edges follow the same cutoff rule and the same equivariant interaction blocks as ordinary edges, and the two copies are supervised through a single joint energy and its derivatives.

3 Background: Training Machine Learning Force Fields

Data. A configuration of N atoms is described by its atomic species $z_1, \dots, z_N$ and its atomic positions $r_1, \dots, r_N \in \mathbb{R}^3$. An electronic-structure calculation labels it with a total energy E^* and with a force $f_i^* \in \mathbb{R}^3$ on each atom i. We write Z, R, E^* and F^* for the species, positions, reference energy and reference forces of one configuration, and $\mathcal{D} = \{(Z^{(s)}, R^{(s)}, E^{*(s)}, F^{*(s)})\}_{s=1}^{M}$ for a dataset of M labeled configurations. We omit the configuration index when it is clear and suppress the dependence on species in energy functions.

Representing a molecular conformation as a graph. A geometric force field represents a conformation as a graph $\mathcal{G}(R)$ whose nodes are atoms. Two distinct atoms are connected by a directed

edge whenever their distance falls below a cutoff radius r_c,

$$\mathcal{E}(R) = \big\{(i,j) : i \neq j,\ \|r_j - r_i\|_2 < r_c\big\}, \qquad \mathcal{N}(i) = \{j : (i,j) \in \mathcal{E}(R)\}. \tag{1}$$

Each edge carries geometric features e_{ij} derived from the atomic positions. Architectures differ in which geometric quantities they use: interatomic distances alone, distances together with directional information such as unit vectors or spherical harmonics, or distances together with angles between adjacent edges and dihedral angles. Each conformation therefore enters training through exactly one graph, which is fixed by its atomic positions and the cutoff.

Message passing and energy readout. Node features are initialized from species embeddings, $h_i^{(0)} = \mathrm{Embed}(z_i)$, and updated over L interaction layers,

$$h_i^{(\ell+1)} = U_\ell\Big(h_i^{(\ell)}, \sum_{j \in \mathcal{N}(i)} M_\ell\big(h_i^{(\ell)}, h_j^{(\ell)}, e_{ij}\big)\Big), \qquad \ell = 0, \dots, L-1, \tag{2}$$

where M_ℓ and U_ℓ are learned message and update functions. A readout maps the final features to site energies, and the total energy is their sum,

$$E_\theta(R) = \sum_{i=1}^{N} \epsilon_i, \qquad \epsilon_i = E^0_{z_i} + \phi\big(h_i^{(L)}\big), \qquad f_i^\theta(R) = -\nabla_{r_i} E_\theta(R), \tag{3}$$

where E^0_z is an atomic reference energy and ϕ is a learned readout, possibly with a fixed scale and shift. Because the edge features are computed from relative positions, E_θ is invariant to global translations of the conformation. Equivariant architectures additionally make E_θ rotation invariant and the forces rotation equivariant. Obtaining forces as the negative gradient of a single scalar energy guarantees an energy-conserving force field.

Training objective. Training minimizes a weighted combination of an energy error, normalized per atom, and a force error, averaged over all $3N$ Cartesian components,

$$\mathcal{L} = \frac{w_E}{N^2}\big(E_\theta(R) - E^*\big)^2 + \frac{w_F}{3N}\sum_{i=1}^{N}\big\|f_i^\theta(R) - f_i^*\big\|_2^2, \tag{4}$$

averaged over minibatches, with weights w_E and w_F. Since the forces are themselves gradients of the energy, minimizing Equation (4) requires differentiating through $\nabla_{r_i} E_\theta$.

4 METHOD

Duplicated-graph training modifies two parts of the pipeline of Section 3 and leaves the rest untouched. It replaces the graph $\mathcal{G}(R)$ of a conformation with a joint graph over the molecule and a rigidly translated duplicate of it (Section 4.1), and it replaces the energy readout and training objective of Equations (3) and (4) with copy-averaged predictions and copy-wise supervision (Section 4.2). The message and update functions, the readout, and the reference labels are unchanged, so duplicated-graph training applies to any force field of this form. At inference the duplicated graph can either be kept, averaging predictions over the two copies and over several translations, or removed by distilling the model into an ordinary-graph student (Section 4.3).

4.1 COUPLED DUPLICATED-GRAPH CONSTRUCTION

The aim is to vary the interaction geometry from which a conformation is predicted, without perturbing the conformation itself. A rigid duplicate achieves this: its placement relative to the molecule varies, while its internal geometry and its atom-wise correspondence to the original are preserved. Each time a labeled conformation is visited during training, we sample one translation $t \sim \mathcal{N}(0, \sigma^2 I_3)$, where σ is the standard deviation of each Cartesian component of t, and place two copies of every atom,

$$r_i^{(0)} = r_i, \qquad r_i^{(1)} = r_i + t, \qquad i = 1, \dots, N. \tag{5}$$

Throughout, the superscript $a \in \{0,1\}$ labels the two copies, with $a = 0$ the original molecule and $a = 1$ its duplicate, and $R^{(a)} = (r_1^{(a)}, \ldots, r_N^{(a)})$ collects the positions of copy a. Species are copied along with positions, so node (i, a) carries species z_i. All duplicate atoms share the same t, which is what makes the duplicate rigid: within either copy, all interatomic distances and angles equal those of the original conformation, so E^* and $f_1^*, \ldots, f_N^*$ remain exact labels for it.

Applying the cutoff rule of Equation (1) once to all $2N$ nodes yields the duplicated graph $\mathcal{G}_{\mathrm{dup}}(R; t)$, whose nodes and edges are

$$\begin{aligned} \mathcal{V}_{\mathrm{dup}} &= \big\{(i,a) : \ i = 1, \ldots, N, \ a \in \{0,1\}\big\}, \\ \mathcal{E}_{\mathrm{dup}}(R;t) &= \big\{\big((i,a),(j,b)\big) : \ (i,a) \neq (j,b), \ \big\|r_j^{(b)} - r_i^{(a)}\big\|_2 < r_c\big\}. \end{aligned} \tag{6}$$

The edge set contains within-copy edges, for which $a = b$, and cross-copy edges, for which $a \neq b$; corresponding atoms of the two copies are connected by the same rule as any other pair. Since $r_j^{(b)} - r_i^{(a)} = (r_j - r_i) + (b - a)\, t$, within-copy geometry reproduces the original conformation exactly, while cross-copy geometry shifts with the sampled translation. Resampling t at every visit therefore presents the same labeled conformation through a family of different interaction geometries.

Message passing proceeds on $\mathcal{G}_{\mathrm{dup}}(R; t)$ exactly as in Equation (2). The two copies share species embeddings, message and update functions, and readout, and cross-copy edges are processed by the same functions as within-copy edges rather than by a separate network. The copies therefore interact throughout message passing instead of being encoded independently; the index a serves only to group corresponding atoms for the readout and supervision below. Validation and test evaluations of the duplicated-graph model retain the duplicated graph and use reproducible, conformation-indexed offsets.

4.2 Energy–Force Readout and Training Objective

Energy and force readout. Applying the readout of Equation (3) to the duplicated graph gives a site energy $\epsilon_{i,a}$ for each of its $2N$ nodes. Their sum splits by copy,

$$E_{\mathrm{joint}}\big(R^{(0)}, R^{(1)}\big) = \sum_{a \in \{0,1\}} \sum_{i=1}^{N} \epsilon_{i,a} = E^{(0)} + E^{(1)}, \qquad E^{(a)} = \sum_{i=1}^{N} \epsilon_{i,a}, \tag{7}$$

where both copy sums depend on the whole duplicated graph through the cross-copy edges. Treating the two sets of positions as independent inputs gives a force $f_i^{(a)} = -\nabla_{r_i^{(a)}} E_{\mathrm{joint}}$ on every node, and the prediction for the labeled conformation averages over the two copies,

$$\widehat{E}(R;t) = \frac{1}{2} E_{\mathrm{joint}}\big(R^{(0)}, R^{(1)}\big), \qquad \widehat{f}_i(R;t) = \frac{1}{2}\big(f_i^{(0)} + f_i^{(1)}\big) = -\nabla_{r_i} \widehat{E}(R;t). \tag{8}$$

The last identity holds for fixed t by the chain rule, since displacing atom i of the conformation displaces both of its copies; Appendix E gives the derivation and the effect of joint centering. Halving also counts the duplicated atomic reference energies once. The averaged prediction is thus the gradient of a single scalar energy, so the duplicated-graph model remains energy conserving in the sense of Section 3. We write $F^{(a)} = (f_1^{(a)}, \ldots, f_N^{(a)})$ for the forces on copy a and $\widehat{F} = (\widehat{f}_1, \ldots, \widehat{f}_N)$ for the averaged forces; the appendices use this notation.

Training objective. Duplicated-graph training keeps the energy term of Equation (4), evaluated at the averaged energy, and replaces its force term by supervising both copies with the same reference forces,

$$\mathcal{L}_{\mathrm{dup}} = \frac{w_E}{N^2}\big(\widehat{E} - E^*\big)^2 + \frac{w_F}{3N(2+\lambda)} \sum_{i=1}^{N} \Big(\big\|f_i^{(0)} - f_i^*\big\|_2^2 + \big\|f_i^{(1)} - f_i^*\big\|_2^2 + \lambda \big\|f_i^{(0)} - f_i^{(1)}\big\|_2^2 \Big), \tag{9}$$

shown for one conformation with unit data weights. Neither copy has a target of its own: both fit the reference forces of the original conformation, and the third term penalizes disagreement between them; the appendix shows that such disagreement is penalized even at $\lambda = 0$. The weights w_E, w_F and $\lambda \geq 0$ are training-protocol choices. Appendix E.3 rewrites the force term of Equation (9) as a fit

of the averaged force $\widehat{f}_i$ to the reference, plus a penalty on the disagreement between the two copies, and states the exact minibatch reduction with configuration and property weights; Appendix E.4 identifies that disagreement with the energy's first-order response to the relative position of the two copies.

4.3 INFERENCE

A model trained on duplicated graphs can be deployed in two ways. It can keep the duplicated graph at inference and average its predictions, or it can be distilled into a student that runs on the ordinary molecular graph. The two options trade accuracy against cost, and the choice depends on whether the duplicated graph is affordable in the target application.

Averaging duplicated-graph predictions. The first option evaluates the model exactly as it was trained. A single duplicated graph already yields the averaged prediction of Equation (8), which combines the two copies at the cost of one forward pass on $2N$ nodes. Accuracy can be improved further by averaging over K translations $t_1, \ldots, t_K$,

$$\widehat{E}_K(R) = \frac{1}{K}\sum_{k=1}^{K} \widehat{E}(R; t_k), \qquad \widehat{f}_{K,i}(R) = \frac{1}{K}\sum_{k=1}^{K} \widehat{f}_i(R; t_k), \tag{10}$$

which requires evaluating K duplicated graphs. Two properties matter in practice. First, the offsets must be held fixed rather than resampled: for any fixed $t_1, \ldots, t_K$ the averaged forces remain the negative gradient of the averaged energy, so the deployed model is still energy conserving, whereas resampling offsets along a trajectory changes the energy surface being integrated (Appendix E.5). We therefore draw the evaluation offsets once and index them by conformation, which also makes evaluation reproducible. Second, averaging improves accuracy while increasing computation; Section 5.5 measures the resulting accuracy and latency at several values of K.

Distilling into an ordinary-graph student. The second option removes the duplicated graph from deployment entirely. We train a student with the same architecture as the baseline force field, operating on the ordinary N-node graph, and supervise it both with the reference energies and forces and with hidden-state targets taken from a duplicated-graph teacher. The student's two interaction blocks are matched to those of the teacher, so it is asked to reproduce the internal representations that the duplicated graph induced, not only the teacher's outputs. At inference, it requires neither duplicated nodes nor cross-copy edges, avoiding the overhead of duplicated-graph evaluation. The distillation protocol and hyperparameters are provided in Appendix B.

The student is trained separately rather than obtained by transforming the duplicated-graph predictor, and on rMD17 it improves on a baseline trained from the same labels while matching or beating its teacher on five of the ten molecules (Section 5). Deployment therefore has a choice: average over displacements on the duplicated graph, or run the student on an ordinary graph.

5 EXPERIMENTS

Our experiments ask three questions. Does training on duplicated graphs improve accuracy on molecules of different size and flexibility? Does the improvement survive a shift to conformations never seen in training? And can it be retained by a model that runs on an ordinary graph at deployment? We answer them on rMD17, 3BPA and MD22, and then isolate the contribution of the duplicated molecular structure against simpler perturbations and measure what the construction costs at inference.

5.1 EXPERIMENT SETUP

Models. All models use MACE (Batatia et al., 2022) as the force-field architecture, whose interaction layers instantiate Equation (2) with higher-order equivariant messages. We compare three models trained from identical labels. MACE is the baseline, trained on the ordinary molecular graph. MACE (Duplicated Graph), abbreviated DupG in tables, is trained on duplicated graphs and evaluated on them, averaging the two copies and, when $K > 1$, averaging over K fixed displacements.

MACE (Distillation), abbreviated Distill in tables, is the student of Section 4.3: it is trained to match the intermediate representations of a duplicated-graph teacher while fitting the same reference labels, and it uses only the ordinary graph at inference.

Benchmarks and metrics. We evaluate ten rMD17 molecules (Christensen & von Lilienfeld, 2020), seven MD22 systems (Chmiela et al., 2023), and the temperature- and dihedral-shifted 3BPA test sets. Each rMD17 and MD22 task uses 950 training configurations, and a separate model is trained per task. We report mean absolute errors (MAE) for rMD17 and MD22 and root mean squared errors (RMSE) for 3BPA. Forces are in meV/Å and energies in meV per configuration. Training schedules, displacement scales and further protocol details are in Appendix A.

5.2 Small molecules: rMD17

Duplicated graphs improve force accuracy on every rMD17 molecule. Table 1 reports all three models. With $K = 20$, the duplicated graph reduces force MAE relative to MACE by between 3.91% on naphthalene and 26.00% on malonaldehyde, with aspirin, ethanol and paracetamol among the larger gains. Because the ten molecules differ in size, flexibility and chemistry, the gain holds across molecular geometries and compositions. Energy errors are reported alongside the forces throughout.

The improvement also survives the removal of the duplicated graph at deployment. Distilling the teacher's intermediate representations into an ordinary-graph student lowers both force and energy MAE below MACE on all ten molecules, by 15.6% on average for force. The student matches or beats its own teacher on five molecules for force and on nine for energy, at the inference cost of a plain MACE model. Part of what the duplicated graph teaches is therefore expressible on the ordinary graph.

Table 1: rMD17 test MAE with 950 training configurations per molecule. E is in meV and F in meV/Å. The duplicated-graph model averages the two copies over $K = 20$ displacements. Best value per column and metric in bold.

Model		Asp.	Azo.	Benz.	Eth.	Malon.	Naph.	Parac.	Salic.	Tol.	Urac.
MACE	E	2.52	0.95	0.14	0.45	0.73	0.40	1.32	0.73	0.40	0.48
	F	7.29	3.02	0.47	3.03	5.19	1.41	5.45	4.12	1.58	2.96
MACE (DupG)	E	**2.05**	0.96	0.26	0.40	0.67	0.55	1.30	0.87	0.43	0.62
	F	5.90	**2.63**	**0.38**	2.28	**3.84**	1.36	**4.22**	**3.52**	**1.51**	2.46
MACE (Distill)	E	2.39	**0.84**	**0.06**	**0.34**	**0.62**	**0.26**	**1.10**	**0.64**	**0.35**	**0.34**
	F	**5.87**	2.77	0.41	**2.23**	3.88	**1.33**	4.26	3.55	**1.51**	**2.42**

5.3 Conformational shift: 3BPA

The gain grows as the test conformations move away from the training distribution. Training uses 300 K conformations only; evaluation covers trajectories at 300, 600 and 1200 K together with a dihedral scan. Table 2 shows lower force RMSE for the duplicated graph on all four sets, and the relative reduction rises from 4.57% at 300 K to 11.22% and then 16.55% as the evaluation temperature increases. The distilled student is the most accurate model at all three temperatures, improving on the duplicated graph by a further 1 to 3%, and the duplicated graph leads on the dihedral scan. Energy RMSE for the duplicated graph falls as well at 600 K, 1200 K and on the dihedral set. This is the behavior one expects if the cross-copy edges expose the model to interaction geometries that its training conformations do not contain. Appendix C.4 resolves the temperature-level result by reference energy and reference-force magnitude.

Table 2: 3BPA test RMSE. Energy E is in meV per configuration and force F in meV/Å. Models are trained on 300 K conformations only. The duplicated-graph model uses $K = 20$. Best value per column and metric in bold.

Model		300 K	600 K	1200 K	Dihedral
MACE	E	**2.96**	12.14	33.01	10.53
	F	9.40	23.14	65.53	15.92
MACE (DupG)	E	3.25	9.83	**26.54**	**8.13**
	F	8.97	20.55	54.68	**15.84**
MACE (Distill)	E	3.09	**9.60**	26.90	8.26
	F	**8.73**	**20.05**	**54.10**	16.16

5.4 LARGER SYSTEMS: MD22

The construction continues to help when the molecules are an order of magnitude larger. Table 3 shows lower force MAE on all seven MD22 systems with $K = 10$ displacement averaging, with reductions between 9.20% and 20.43%; the largest gains are on Ac-Ala3-NHMe and DHA. Taken with rMD17 and 3BPA, force accuracy improves from small molecules to systems of hundreds of atoms.

Table 3: MD22 test MAE with 950 training configurations. Energy E is in meV and force F in meV/Å. The duplicated-graph model averages $K = 10$ displacements. Best value per column and metric in bold.

Model		Ac-Ala3	DHA	AT-AT	Stachyose	AT-AT-CG-CG	Buckyball	Nanotube
MACE	E	**12.48**	**23.71**	32.46	39.41	**61.60**	21.16	164.65
	F	9.20	7.32	6.76	9.26	6.63	3.40	10.62
MACE (DupG)	E	13.82	24.74	32.62	41.88	65.13	21.08	164.02
	F	7.32	**5.87**	6.00	**7.71**	6.02	2.75	**8.97**
MACE (Distill)	E	13.52	24.67	**31.27**	**35.10**	64.24	**20.90**	**163.86**
	F	**7.29**	5.91	**5.95**	7.75	**5.80**	**2.62**	9.12

5.5 WHAT THE DUPLICATE CONTRIBUTES, AND WHAT IT COSTS

The duplicated molecular structure, not coordinate noise. Two simpler perturbations fail to reproduce the gain on rMD17. Gaussian test-time augmentation perturbs the coordinates of a trained MACE model and averages $K = 20$ predictions; it increases force error on all ten molecules, by 57% on aspirin and by an order of magnitude on the molecules the baseline predicts most accurately. A sparse auxiliary-node design, which adds a few nodes near the molecule instead of a full translated copy, improves force MAE over its MACE reference on nine of ten molecules but stays behind the duplicated graph on every molecule compared. Averaging predictions over perturbed coordinates is therefore not the mechanism, and neither is merely enlarging the graph: what matters is that the added nodes carry the complete molecular geometry. Per-molecule values are in Appendices C.2 and C.3, the auxiliary-node construction in Appendix D, and the sensitivity of the duplicated graph to the translation scale in Appendix C.1.

Accuracy against inference cost. Averaging several displacements improves force accuracy with little additional latency at small K. Averaged over four rMD17 molecules, $K = 5$ reduces mean force MAE by 22% relative to MACE for a 3% increase in median H200 latency, while $K = 20$ buys a further 1% in accuracy at 1.6 times the baseline latency (Table 4). The first few displacements thus capture most of the force improvement; larger K offers a further accuracy–latency trade-off, while distillation provides ordinary-graph inference.

Table 4: Displacement-averaging accuracy and cost on four rMD17 molecules. Latency is the mean of per-molecule median H200 model inference times (ms). Energy and force are test MAE (meV and meV/Å), averaged equally across molecules. The duplicated-graph model averages the two copies and K displacements; the baseline evaluates one ordinary graph. Appendix A gives the timing protocol.

	MACE	MACE (DupG)				
Displacements K	—	1	2	5	10	20
Latency (ms)	19.75	19.89	20.18	20.28	21.76	32.56
Energy MAE	0.45	0.49	0.44	0.41	0.39	0.38
Force MAE	2.91	2.43	2.33	2.27	2.25	2.24

6 Discussion and limitations

Why the duplicate helps. Our results are consistent with a simple reading of the construction. A labeled conformation constrains the model only through the interaction geometries that its own cutoff graph contains, and those are fixed once the coordinates are given. The duplicate leaves the conformation, the labels and the network untouched, but surrounds each atom with a second, shifted image of the molecule, so the same label now constrains the model on a family of interaction geometries indexed by the translation. This reading predicts the pattern we observe. The benefit should be largest where the training conformations cover interaction geometries poorly, which is what the 3BPA results show: the relative force improvement grows monotonically as the evaluation temperature moves away from the 300 K training set, and is smallest on the dihedral scan, whose local environments most resemble the training data. It also explains why the distilled student recovers much of the gain. If the duplicate mainly shapes which function the network settles on, rather than supplying information that must be present at test time, then a student trained to reproduce the teacher's internal representations can inherit that function while evaluating an ordinary graph.

Costs of the two inference routes, and of training. The gains are not free, and the two inference routes trade accuracy against latency in opposite directions. A single duplicated graph recovers most of the force improvement at close to baseline latency, while the accuracy left beyond five displacements is small and costs roughly 1.6 times the baseline latency at $K = 20$ (Table 4); the offsets must also stay fixed for the averaged model to remain energy conserving. The distilled student matches baseline latency but is separately trained, so its accuracy has to be established per benchmark. Training is more expensive too: a duplicated graph carries twice the nodes and more than twice the edges, so each update costs more time and memory, our comparisons match optimizer steps rather than wall-clock time, and the student additionally requires training its teacher first. The construction therefore spends compute at training time, and optionally at inference time, to buy accuracy from a fixed set of expensive labels.

7 Conclusion

We introduced duplicated-graph training, which presents each labeled conformation to a force field together with a rigidly translated copy of itself in one cutoff graph: the copy is rigid, so every intramolecular distance and angle is preserved and the reference labels stay exact, while the cross-copy edges give that single label a family of interaction geometries to constrain. Applied to MACE, it lowers force error on all ten rMD17 molecules, all seven MD22 systems and all four 3BPA test sets, with the gain under conformational shift growing from 4.57% at 300 K to 16.55% at 1200 K, and distillation carries the improvement to a student that runs on an ordinary graph at baseline cost. The result suggests that the graph through which a labeled conformation reaches the network is a design choice rather than a given, and that varying it is a cheap source of supervision orthogonal to better architectures, better sampling and larger pretrained models.

AI use statement

In this work, we used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims,

assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, and interpret results. Generating synthetic data sets, assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, draft parts of a research paper, summarise or analyse existing literature, discover research topics or identify gaps, brainstorm, source or search for information, edit a research paper to improve readability, identify relevant literature, format references, suggest a structure for a research paper, and propose a title or keywords for a research paper. We have reviewed all AI-assisted work. LLM-generated code was verified and tested by the authors before the runs it produced were used, every number reported here is the output of an executed run rather than a model's account of one, and every reference was checked against the publisher record. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Every component needed to reproduce the reported results is specified in the paper. The duplicated-graph construction, the joint readout and the training objective are defined in Section 4, and the inference procedures in Section 4.3; the standard training setup they modify is stated in Section 3. Section 5.1 names the benchmarks, the architecture and the metrics. Appendix A gives the data and evaluation protocol, including how every train, validation and test split is formed from the public rMD17, MD22 and 3BPA releases, the per-task translation scales, and the complete architecture, optimization and model-selection settings of the baseline and duplicated-graph runs; Appendix B does the same for the student, together with the feature-matching objective and the layers it is applied to. Appendix A states the hardware and timing procedure behind the reported latencies, and Appendix D the construction used in the auxiliary-node comparison. The assumptions behind each analytical claim in the main text, and the derivations that support them, are given in Appendix E. Source code, configuration files and the job launchers for each experiment are withheld during review for anonymity and will be released with the paper.

References

Uri Alon and Eran Yahav. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*, 2021. URL https://iclr.cc/virtual/2021/poster/2575.

Mehdi Azabou, Venkataramana Ganesh, Shantanu Thakoor, Chi-Heng Lin, Lakshmi Sathidevi, Ran Liu, Michal Valko, Petar Veličković, and Eva L Dyer. Half-Hop: A graph upsampling approach for slowing down message passing. In *Proceedings of the International Conference on Machine Learning*, volume 202, pp. 1341–1360, 2023. URL https://proceedings.mlr.press/v202/azabou23a.html.

Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2206.07697.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid Péter Kovács, Janosh Riebesell, et al. A foundation model for atomistic materials chemistry. *The Journal of Chemical Physics*, 163(18):184110, 2025. URL https://arxiv.org/abs/2401.00096.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, 2022. doi: 10.1038/s41467-022-29939-5.

Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98(14):146401, 2007. doi: 10.1103/PhysRevLett.98.146401.

Alessandro Caruso, Jacopo Venturin, Lorenzo Giambagli, Edoardo Rolando, Zakariya El-Machachi, Frank Noé, and Cecilia Clementi. Extending the range of graph neural networks with global encodings. *Nature Communications*, 17(1):1855, 2026. doi: 10.1038/s41467-026-69715-3.

Stefan Chmiela, Valentin Vassilev-Galindo, Oliver T. Unke, Adil Kabylda, Huziel E. Sauceda, Alexandre Tkatchenko, and Klaus-Robert Müller. Accurate global machine learning force fields for molecules with hundreds of atoms. *Science Advances*, 9(2):eadf0873, 2023. doi: 10.1126/sciadv.adf0873.

Anders Christensen and O Anatole von Lilienfeld. Revised md17 dataset. *Materials Cloud Archive*, 7, 2020.

April M. Cooper, Johannes Kästner, Alexander Urban, and Nongnuch Artrith. Efficient training of ANN potentials by including atomic forces via Taylor expansion and application to water and a transition-metal oxide. *npj Computational Materials*, 6:54, 2020. doi: 10.1038/s41524-020-0323-8.

Shikun Feng, Yuyan Ni, Yanyan Lan, Zhi-Ming Ma, and Wei-Ying Ma. Fractional Denoising for 3D Molecular Pre-training. In *Proceedings of the International Conference on Machine Learning*, volume 202, pp. 9938–9961, 2023. URL https://proceedings.mlr.press/v202/feng23c.html.

Jonathan Godwin, Michael Schaarschmidt, Alexander Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple GNN Regularisation for 3D Molecular Property Prediction & Beyond. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1wVvweK3oIb.

Zhichun Guo, Tong Zhao, Yozen Liu, Kaiwen Dong, William Shiao, Mingxuan Ju, Neil Shah, and Nitesh V. Chawla. Node Duplication Improves Cold-start Link Prediction. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=hIOTzz87N9.

Arthur Kosmala, Johannes Gasteiger, Nicholas Gao, and Stephan Günnemann. Ewald-based Long-Range Message Passing for Molecular Graphs. In *Proceedings of the International Conference on Machine Learning*, volume 202, 2023. URL https://proceedings.mlr.press/v202/kosmala23a.html.

Dávid Péter Kovács, J. Harry Moore, Nicholas J. Browning, Ilyes Batatia, Joshua T. Horton, Yixuan Pu, Venkat Kapil, William C. Witt, Ioan-Bogdan Magdău, Daniel J. Cole, and Gábor Csányi. MACE-OFF: Short-Range Transferable Machine Learning Force Fields for Organic Molecules. *Journal of the American Chemical Society*, 147(21):17598–17611, 2025. doi: 10.1021/jacs.4c07099.

Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph Matching Networks for Learning the Similarity of Graph Structured Objects. In *Proceedings of the International Conference on Machine Learning*, volume 97, pp. 3835–3845, 2019. URL https://proceedings.mlr.press/v97/li19d.html.

Yunyang Li, Yusong Wang, Lin Huang, Han Yang, Xinran Wei, Jia Zhang, Tong Wang, Zun Wang, Bin Shao, and Tie-Yan Liu. Long-Short-Range Message-Passing: A Physics-Informed Framework to Capture Non-Local Interaction for Scalable Molecular Dynamics Simulation. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2304.13542.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. R-Drop: Regularized Dropout for Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10890–10905, 2021. URL https://arxiv.org/abs/2106.14448.

Yi-Lun Liao, Tess Smidt, Muhammed Shuaibi, and Abhishek Das. Generalizing Denoising to Non-Equilibrium Structures Improves Equivariant Force Fields. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=whGzYUbIWA.

Yuyan Ni, Shikun Feng, Wei-Ying Ma, Zhi-Ming Ma, and Yanyan Lan. Sliced Denoising: A Physics-Informed Molecular Pre-Training Method. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=liKkG1zcWq.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkx1qkrKPr.

Makoto Takamoto, Viktor Zaverkin, and Mathias Niepert. Physics-Informed Weakly Supervised Learning For Interatomic Potentials. In *Proceedings of the International Conference on Machine Learning*, volume 267, pp. 58282–58310, 2025. URL https://proceedings.mlr.press/v267/takamoto25a.html.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A.

Oliver T. Unke, Stefan Chmiela, Huziel E. Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T. Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Machine Learning Force Fields. *Chemical Reviews*, 121(16):10142–10186, 2021. doi: 10.1021/acs.chemrev.0c01111.

Yusong Wang, Chaoran Cheng, Shaoning Li, Yuxuan Ren, Bin Shao, Ge Liu, Pheng-Ann Heng, and Nanning Zheng. Neural P^3M: A Long-Range Interaction Modeling Enhancer for Geometric GNNs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d9c7c8bd6ad4cebb7d006e5109e0b682-Abstract-Conference.html.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022. doi: 10.1038/s42256-022-00447-x.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5812–5823, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe230348e9a12c13120749e3f9fa4cd-Abstract.html.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via Denoising for Molecular Property Prediction. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tYIMtogyee.

Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and Weinan E. DP-GEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253:107206, 2020. doi: 10.1016/j.cpc.2020.107206.

A EXPERIMENTAL DETAILS

Data and evaluation protocol. For rMD17 we use the official first train/test fold for each molecule: 1,000 configurations are reserved for testing, while a random split of the official 1,000-frame training fold gives 950 training and 50 validation configurations. Each MD22 system uses 950 training and 50 validation configurations. All seven MD22 systems have fixed test sets that are disjoint from the training and validation sets and held out during model selection. For Ac-Ala3-NHMe and DHA, we sample the 1,000 training and validation frames from a previously defined pool and retain its established test set. For the other five systems, a random permutation of the complete trajectory assigns 950 frames to training, 50 to validation, and the remaining frames to a fixed test set, whose size Table A1 reports. The 3BPA models use 450 training and 50 validation configurations drawn from the 300 K set and are evaluated on the temperature-specific and dihedral test sets. One model is trained per task. Errors are computed on the physical atoms only, and energy errors are reported per configuration rather than per atom. For a duplicated graph, the two copies are averaged first and, when $K > 1$, the resulting predictions are averaged over the K fixed displacements; all metrics are computed after this prediction averaging. Evaluation uses $K = 20$ displacements on rMD17 and 3BPA and $K = 10$ on MD22, as listed in Table A2.

Duplicated-graph training. Table A2 collects the architecture, graph, optimization and model-selection settings of the baseline and duplicated-graph models on all three benchmarks. Baselines and duplicated-graph models share the data split, architecture and schedule of their benchmark and differ only in the graph they are trained and evaluated on, and in the duplicate-consistency coefficient λ of Equation (9), which is zero for every baseline. Checkpoints are selected separately for each model using its own validation loss, evaluated on the graph that model uses, so duplicated-graph validation applies a fixed validation displacement. No test fold is used for selection. The translation scale σ varies by molecule for rMD17 and MD22 and is listed, with the MD22 test-set sizes, in Table A1; the 3BPA runs use 0.10 Å. Where the table reports a decay factor and a patience, the learning rate is multiplied by that factor once the given number of epochs or validation evaluations passes without improvement, and early stopping applies the same rule to training. The rMD17 and MD22 schedules use the two training stages of the reference MACE implementation. Stage 1 starts at epoch 1 and Stage 2 at the epoch given in the table, which counts from the same origin as the maximum epoch count above it, so rMD17 spends its first 2,000 epochs in Stage 1 and the remainder in Stage 2. At that epoch the learning rate is reset to the Stage 2 value and the energy-to-force weight ratio is replaced by the Stage 2 ratio, while training continues from the weights reached so far. The 3BPA runs use Stage 1 only, which is what the dashes in that column mark. Where both stages run, their checkpoints are compared on validation data before the test set is evaluated.

Latency measurement. We measure the GPU time needed to predict energy and forces for one conformation on a single NVIDIA H200, taking the median request latency per system over three rounds of 32 fixed test configurations after five warm-up calls, and then the mean across systems reported in Table 4. CUDA events enclose the complete model call and the device is synchronized before the elapsed time is recorded, so the measurement covers both the energy forward pass and the autograd computation of forces. Neighbor graphs are built on the CPU beforehand and are not timed. For $K > 1$ the displaced duplicated graphs are packed as separate connected components of one batch and evaluated in a single model call, so the displacements run in parallel on the device.

Table A1: Per-task settings: the translation scale σ (Å) selected on validation data, and the size of each MD22 test set. Every rMD17 test set holds 1,000 configurations.

rMD17	Asp.	Azo.	Benz.	Eth.	Malon.	Naph.	Parac.	Salic.	Tol.	Urac.
σ	0.20	0.05	0.05	0.10	0.15	0.07	0.07	0.07	0.07	0.10

MD22	Ac-Ala3	DHA	AT-AT	Stachyose	AT-AT-CG-CG	Buckyball	Nanotube
σ	0.10	0.10	0.10	0.10	0.10	0.01	0.10
Test frames	79,109	61,753	19,001	26,272	9,153	5,102	4,032

Table A2: Baseline and duplicated-graph training configuration. Both models of a benchmark share every setting except the graph they run on and the loss weights where indicated. A dash marks a setting that does not apply.

Setting	rMD17	MD22	3BPA
Data			
Training configurations	950	950	450
Validation configurations	50	50	50
Architecture and graph			
Interaction blocks	2	2	2
Hidden channels ($0e/1o/2e$)	$256/256/256$	$256/256/256$	$256/256/256$
Correlation order	3	3	3
Radial basis functions	8	8	8
Cutoff (Å)	6	5	5
Displacements K at evaluation	20	10	20
Optimization			
Optimizer	Adam (AMSGrad)	Adam (AMSGrad)	Adam (AMSGrad)
Initial learning rate	0.01	$\{0.005,\ 0.01\}$	0.01
Weight decay	5×10^{-7}	5×10^{-7}	5×10^{-7}
EMA decay	0.99	0.99	0.99
Optimizer updates (max)	665,000	665,000	665,000
Epochs (max)	3,500	1,400	7,389
Updates per epoch	190	475	90
Loss weights			
Stage 1, E:F, baseline	9:1000	10:1000	27:1000
Stage 1, E:F, duplicated graph	9:1000	10:1000	9:1000
Stage 2, E:F	9:1000	1000:10	—
Consistency coefficient λ (DupG)	1	1	0
Schedule			
Stage 2 starts at epoch	2,000	970	—
Stage 2 learning rate	0.001	0.001	—
Validation interval (epochs)	1	18	1 (baseline), 2 (DupG)
Learning-rate decay factor	0.8	0.8	0.8
Decay patience (epochs)	50	30	50
Early-stopping patience (evaluations)	100	5	—

B DISTILLATION DETAILS

The teacher is a trained duplicated-graph model, and the student is initialized from its weights but evaluates an ordinary N-atom graph containing neither duplicate nodes nor cross-copy edges. We match the outputs of the two corresponding MACE interaction blocks, after the product operation and before the energy readout. For each physical atom, the original and duplicate teacher nodes are paired by atom index to form a single cached feature target, so teacher and student have one aligned feature vector per physical atom at each matched block, of equal dimension and requiring no learned projection.

Let $h^{S,\ell}_{i,b}$ and $\bar{h}^{T,\ell}_{i,b}$ denote the student feature and cached teacher target for physical atom i, interaction block $\ell \in \{1, 2\}$, and irreducible-representation block b. We divide each squared difference by the squared RMS $s^2_{\ell,b}$ of the corresponding teacher-target block, computed on the training set. The feature objective is

$$\mathcal{L}_{\text{feat}} = \frac{1}{2} \sum_{\ell=1}^{2} \frac{1}{B_\ell} \sum_{b=1}^{B_\ell} \text{mean}_{i,c} \left[\left(\frac{h^{S,\ell}_{i,b,c} - \bar{h}^{T,\ell}_{i,b,c}}{s_{\ell,b}} \right)^2 \right], \tag{11}$$

where B_ℓ is the number of representation blocks and c indexes their components. The student minimizes $w_E\mathcal{L}_E + w_F\mathcal{L}_F + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}}$, where $\mathcal{L}_E$ and $\mathcal{L}_F$ use the reference energies and forces. The teacher supplies hidden-feature targets only, not an additional energy or force prediction loss. Table A3 gives the student configuration. Learning rates warm up linearly from one tenth of the

Table A3: Ordinary-graph student configuration. A dash marks a setting that does not apply to that benchmark.

Setting	rMD17	MD22	3BPA
Objective			
Matched interaction blocks	2	2	2
w_E:w_F	9:1000	10:1000	27:1000
Feature weight λ_{feat}	$\{0.1,\ 1,\ 10\}$	$\{0.1,\ 1\}$	1
Optimization			
Optimizer	Adam (AMSGrad)	Adam (AMSGrad)	Adam (AMSGrad)
EMA decay	0.99	0.99	0.99
Weight decay	5×10^{-7}	5×10^{-7}	5×10^{-7}
Batch size	5	1–2	5
Effective batch size	5	2	5
Peak learning rate	10^{-4}	10^{-4}	10^{-4}
Warm-up updates	2,000	2,000	2,000
Learning-rate decay factor	0.8	0.8	0.8
Decay patience (evaluations)	50	50	50
Learning-rate floor	10^{-6}	10^{-6}	10^{-6}
Gradient clipping (norm)	10	10	None
Stopping			
Validation interval (epochs)	1	1	2
Patience (epochs)	100	100	—
Patience (evaluations)	—	—	200
Relative improvement threshold	—	—	0.1%

peak value over the warm-up updates listed, and then decay by the given factor whenever the stated number of validation evaluations passes without improvement. For rMD17 and MD22, training stops after the listed patience in non-improving epochs, once the learning rate has decreased at least once; the 3BPA student stops after the listed number of evaluations without a relative improvement above the stated threshold, once its learning rate has fallen to one fifth of the peak value. In every case we select the moving-average checkpoint with the lowest validation force error, and at inference the student evaluates a single ordinary graph.

C Supplementary results

C.1 Sensitivity to the translation scale

Figure A1 varies σ on six rMD17 molecules. Scales at or below 0.003 Å are worse than the baseline on all six, the intermediate range from 0.03 to 0.1 Å is uniformly better, and the largest scales are worse on some molecules, so the working range is bounded on both sides.

C.2 Coordinate perturbation at test time

Holding the baseline weights fixed, we add zero-mean coordinate noise of standard deviation 0.001 Å to the physical atoms and average $K = 20$ predictions. This is an inference-only control, not a noisy training run. Table A4 shows that it is worse than the unperturbed baseline on every molecule, by a wide margin on the ones the baseline predicts most accurately.

Table A4: rMD17 test force MAE (meV/Å) for the baseline and for Gaussian test-time augmentation with $K = 20$.

	Asp.	Azo.	Benz.	Eth.	Malon.	Naph.	Parac.	Salic.	Tol.	Urac.
MACE	**7.29**	**3.02**	**0.47**	**3.03**	**5.19**	**1.41**	**5.45**	**4.12**	**1.58**	**2.96**
Perturbed	11.48	8.99	7.96	7.23	9.98	7.90	9.95	9.19	7.71	9.47

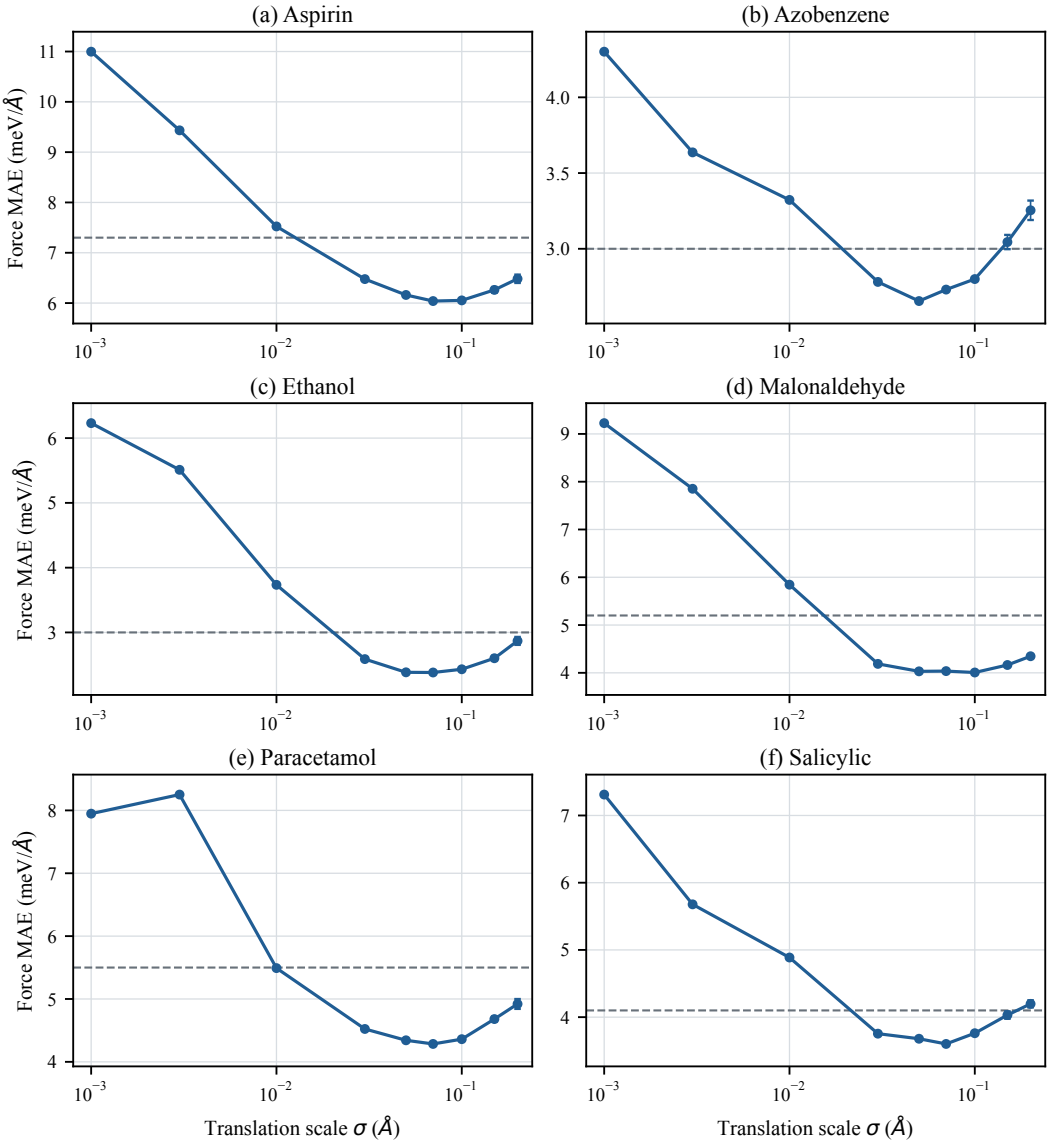

Figure A1: rMD17 force error against translation scale σ. Curves show the mean of 20 displacement-wise copy-averaged force MAEs and error bars their standard deviation; dashed lines mark the baseline force MAE. Table 1 instead averages predictions over the 20 displacements before computing the MAE, which is why its values are lower.

C.3 Sparse auxiliary nodes

A weaker form of graph expansion places a few auxiliary nodes close to a fraction of the atoms instead of adding a translated copy of the molecule; Appendix D gives the construction. Table A5 reports, for each molecule, the configuration with the lowest force error among those we tried, given by the fraction of anchored atoms and the placement distance. Force error improves on nine of ten molecules but by far less than the duplicated graph achieves, while energy error changes in both directions.

Table A5: rMD17 sparse auxiliary-node control. Ratio is the fraction of anchored atoms and d the placement distance; E is in meV and F in meV/Å. The MACE rows reproduce the corresponding results in Table 1; Aux. rows use auxiliary nodes during training and evaluation.

	Asp.	Azo.	Benz.	Eth.	Malon.	Naph.	Parac.	Salic.	Tol.	Urac.
Ratio	0.50	0.25	0.25	0.25	0.25	0.25	0.50	0.25	0.25	0.25
d (Å)	0.005	0.005	0.010	0.005	0.005	0.010	0.005	0.010	0.005	0.010
MACE E	**2.52**	0.95	0.14	**0.45**	0.73	**0.40**	1.32	**0.73**	0.40	0.48
Aux. E	2.60	**0.79**	**0.13**	0.47	**0.71**	0.42	**1.19**	0.79	**0.31**	**0.48**
MACE F	7.29	3.02	0.47	3.03	5.19	1.41	5.45	4.12	**1.58**	2.96
Aux. F	**6.97**	**2.78**	**0.44**	**2.85**	**4.73**	**1.39**	**5.15**	**4.07**	1.66	**2.89**

Table A6: 3BPA pooled component-force RMSE (meV/Å) by reference-energy region. Relative reductions use Equation 12; the $K = 20$ column averages the fixed displacement predictions before scoring. The four exceedance groups partition the row above the training maximum.

Region	n	MACE	Duplicated graph		Relative reduction (%)	
			$K = 1$	$K = 20$	$K = 1$	$K = 20$
$E \leq Q_{75}$	1,198	8.716	8.654	8.394	0.714	3.698
$Q_{75} < E \leq Q_{95}$	370	10.616	10.405	10.075	1.991	5.094
$Q_{95} < E \leq E_{\max}$	418	15.914	14.729	14.351	7.444	9.818
$E > E_{\max}$	3,960	50.847	43.665	42.712	14.124	15.999
Exceedance group 1	990	21.192	19.416	19.017	8.379	10.260
Exceedance group 2	990	28.785	25.934	25.328	9.904	12.011
Exceedance group 3	990	52.322	44.946	44.071	14.097	15.770
Exceedance group 4	990	79.538	67.504	65.968	15.130	17.061

C.4 Force-error reductions across and beyond the training-energy range

The temperature comparison in Table 2 aggregates conformations with different physical states. Here we locate the force-error reduction within those states by comparing the frozen MACE baseline and duplicated-graph model on the same pooled 300, 600 and 1200 K test conformations. Groups are defined by DFT reference energy relative to quantiles and the maximum of the union of the models' training configurations. For a group B, we pool errors over its configurations, physical atoms and Cartesian components, and report

$$\Delta_{\mathrm{rel}}(B) = 100 \left(1 - \frac{\mathrm{RMSE}_{\mathrm{dup}}(B)}{\mathrm{RMSE}_{\mathrm{MACE}}(B)} \right) \%. \tag{12}$$

This is a ratio of pooled force root mean squared errors, not an average of per-configuration percentages. The single-displacement duplicated-graph prediction first averages the forces of the original and translated copies of one joint graph.

Reference-energy strata. Figure A2 shows that the relative force-error reduction grows across the four reference-energy regions: 0.714% at or below the training-union 75th percentile, 1.991% between the 75th and 95th percentiles, 7.444% between the 95th percentile and the observed training maximum, and 14.124% above that maximum. The last region contains 3,960 conformations. Splitting it into four frozen, equal-count groups of 990 configurations ordered by energy exceedance gives reductions of 8.379%, 9.904%, 14.097% and 15.130%. The increase therefore continues within the above-maximum region, rather than arising solely at the observed training-energy boundary. Table A6 gives the underlying pooled errors.

Reference-force magnitude and absolute gains. Figure A3 stratifies the same frozen predictions by reference energy and reference-force root mean square. Its vertical axes report the mean reduction in per-configuration force RMSE (meV/Å), a different aggregation from Equation 12. The energy and force deciles both show larger absolute reductions toward their upper ranges. The joint energy–force quartiles locate the strongest mean reduction, 10.151 meV/Å among 1,211 conformations,

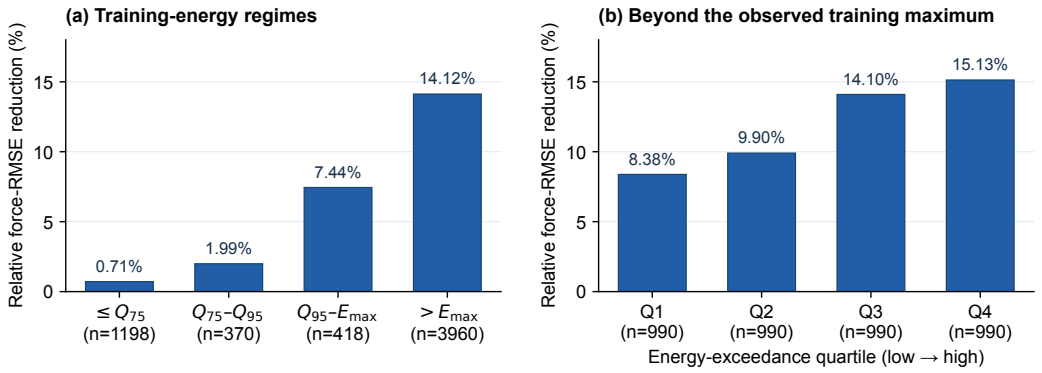

Figure A2: Relative force-RMSE reduction of the single-displacement duplicated-graph model against the MACE baseline on pooled 300, 600 and 1200 K 3BPA configurations. (a) Groups defined by the reference-energy 75th percentile, 95th percentile and observed maximum of the training-configuration union. (b) The 3,960 configurations above that maximum, divided by energy exceedance into four equal-count groups of 990. Both panels use the same zero-based vertical scale. Positive values favor the duplicated-graph model.

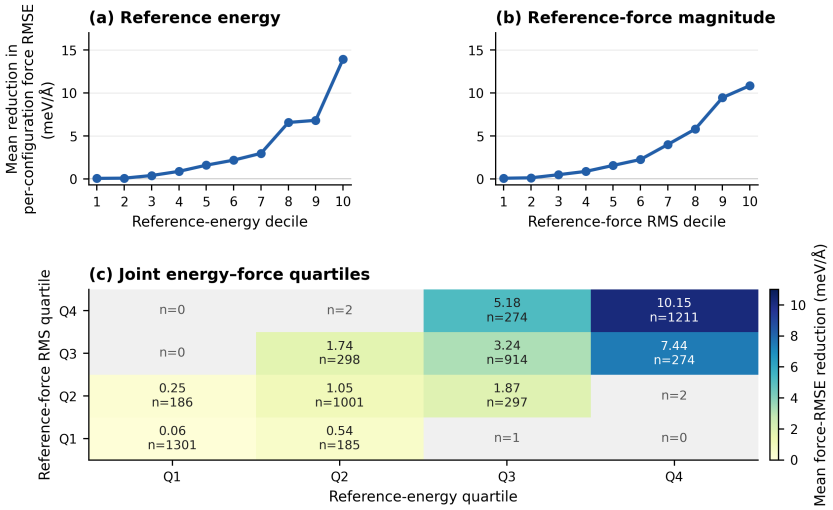

Figure A3: Absolute force-error reductions of the single-displacement duplicated-graph model against MACE. Panels (a) and (b) group pooled test conformations into equal-count reference-energy and reference-force deciles; panel (c) crosses their quartiles. Each value is the mean per-configuration force-RMSE reduction in meV/Å, rather than the relative reduction in Figure A2. Gray cells contain fewer than 30 configurations and display counts but no gain.

where both descriptors are in their highest quartiles. This pattern is also visible within the 1200 K trajectory: the highest-minus-lowest decile difference in mean per-configuration gain is 13.641 meV/Å for reference energy and 7.027 meV/Å for reference-force root mean square. Thus, the physical-state pattern is not exhausted by differences among the three temperature groups.

Multiple displacements. Averaging predictions over $K = 20$ fixed displacements retains the same ordering across reference-energy strata (Table A6). Its relative reductions are 3.698%, 5.094%, 9.818% and 15.999% across the four regions in Figure A2(a), and 10.260%, 12.011%, 15.770% and 17.061% across the above-maximum groups. Figure A4 gives the corresponding absolute-gain

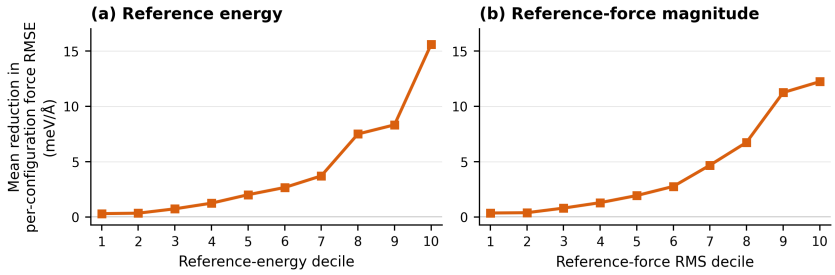

Figure A4: Mean per-configuration force-RMSE reduction (meV/Å) with $K = 20$ displacement averaging, across the same pooled reference-energy and reference-force deciles as Figure A3. The plotted quantity is an absolute gain, not a groupwise relative percentage.

deciles. Displacement averaging improves the predictions, while preserving the concentration of larger gains in the high-energy and strong-reference-force regions.

Conclusion. The duplicated-graph model reduces force error unevenly across the 3BPA configurations studied here. Its relative reduction rises from 0.714% in the lowest training-energy region to 14.124% above the observed training maximum, and increases further across equal-count groups within that upper region. Independent stratification by reference-force magnitude and the within-1200 K contrasts place the larger absolute gains in high-energy, strong-force configurations. The $K = 20$ readout follows the same physical-state pattern. These results identify where the force benefit is concentrated rather than attributing it to temperature alone.

D SPARSE AUXILIARY-NODE CONSTRUCTION

This is the construction behind the sparse auxiliary-node control of Appendix C.3, a weaker form of graph expansion than a complete molecular duplicate. Nodes are placed at a prescribed radius in random directions around sampled real-atom anchors. Real and auxiliary nodes enter the same cutoff graph and participate in message passing. Energy is read out only from real atoms, and force supervision and metrics are restricted to real atoms. Auxiliary nodes can nevertheless affect predictions by changing real-node features during message passing; the absence of a force target does not impose a zero-force condition on them. It therefore probes whether adding a few nearby nodes suffices, rather than isolating one component of the duplicated graph.

Sparse anchor construction. The fixed-distance variant augments N real atoms with at most

$$n_{\max} = \min\left(N, \max\left(1, \left\lfloor \rho N + \frac{1}{2} \right\rfloor\right)\right) \tag{13}$$

auxiliary nodes, for ratio $0 < \rho \leq 1$. Anchors are sampled uniformly without replacement from the eligible atoms. For each anchor a_j, sample $g_j \sim \mathcal{N}(0, I_3)$ and set

$$\xi_j = \frac{g_j}{\|g_j\|_2}, \qquad x_j = r_{a_j} + d\xi_j. \tag{14}$$

Candidates closer than the collision threshold to any non-anchor real atom or previously accepted auxiliary node are rejected. The anchor itself is exempt from that test, allowing microdistance placements. Sampling also requires a cutoff connection and has a bounded retry count. A single neighbor search over all real and auxiliary coordinates supplies real–real, real–auxiliary, and auxiliary–auxiliary edges whenever their distances satisfy the cutoff. The anchor relation does not force an edge outside this rule.

Features, energy masking, and supervision. The selected zero-base variant gives auxiliary nodes an all-zero species vector and appends a scalar indicator (one for auxiliary nodes, zero for real atoms) before the learned linear embedding. Thus zero-base refers to the species input; the projected indicator can yield nonzero hidden features. The auxiliary indicator also defines masks for the

readout and loss. Both atomic reference contributions and learned site energies, including the scale-and-shift contribution, are zeroed on auxiliary nodes. The molecular energy sums only real-node site energies, whose values can still depend on auxiliary-node messages. Force MSE includes only real atoms and energy normalization uses N. Zero entries stored in auxiliary force-label rows are padding, not zero-force supervision. Auxiliary coordinates are treated as independent inputs, so no coordinate gradient flows through $x_j = r_{a_j} + d\xi_j$ back to its anchor. This masking convention is distinct from the duplicated-graph copy objective and its averaged-force identity.

Annealing the auxiliary-node count. Let $p = k/T$ be optimizer-step progress, with hold fraction h and zero-count fraction z, where $0 \le h < z \le 1$. Cosine annealing uses

$$a(p) = \begin{cases} 1, & p \le h, \\ \frac{1}{2}\left[1 + \cos\left(\pi \frac{p-h}{z-h}\right)\right], & h < p < z, \\ 0, & p \ge z. \end{cases} \tag{15}$$

For $\mu_k = n_{\max} a(k/T)$, each configuration receives $n_k = \lfloor \mu_k \rfloor + \mathrm{Bernoulli}(\mu_k - \lfloor \mu_k \rfloor)$ nodes. The rounding and layout draws are reproducible within a run and may change across training visits. Constant-count and hard-switch controls use $a(p) = 1$ and $a(p) = \mathbf{1}[p < p_{\mathrm{switch}}]$. The annealed runs use $d = 0.005$ Å, $h = 0.2$ and $z = 2000/3500$, with a molecule-specific ρ; after $p \ge z$ training continues on ordinary graphs, which are also used for validation and test throughout. This schedule reduces the node count, not the placement distance, and remains a design variant rather than a second primary method.

E Derivations

The identities below use the readout and objective of Section 4. They hold at differentiable configurations, with the translation held fixed when derivatives are taken; species and parameters are suppressed.

E.1 Joint energy and the molecular force

Let $\epsilon_{i,a}$ be the site energy of atom i in copy a, including its atomic reference contribution and learned contribution. The copy sums are $E^{(0)} = \sum_i \epsilon_{i,0}$ and $E^{(1)} = \sum_i \epsilon_{i,1}$, and $E_{\mathrm{joint}} = E^{(0)} + E^{(1)}$. Although grouped by copy, both sums depend on both copies through the cross-copy edges. The reported energy is $\widehat{E} = E_{\mathrm{joint}}/2$. If $e_0(Z_i)$ denotes an atomic reference energy, the reference part of this readout is

$$\frac{1}{2}\left[\sum_{i=1}^{N} e_0(Z_i) + \sum_{i=1}^{N} e_0(Z_i)\right] = \sum_{i=1}^{N} e_0(Z_i). \tag{16}$$

This counting identity does not assume that the learned contributions of the two copies are independent.

For $R^{(0)} = R$ and $R^{(1)} = R + \mathbf{1}_N t^\top$, define $F^{(0)} = -\nabla_{R^{(0)}} E_{\mathrm{joint}}$ and $F^{(1)} = -\nabla_{R^{(1)}} E_{\mathrm{joint}}$. The partial derivatives treat $R^{(0)}$ and $R^{(1)}$ as independent arrays. Differentiating the molecular energy as a function of the original coordinates subsequently applies the duplicate construction:

$$\begin{aligned} -\nabla_R \widehat{E}(R;t) &= -\frac{1}{2}\nabla_R E_{\mathrm{joint}}(R, R + \mathbf{1}_N t^\top) \\ &= -\frac{1}{2}\left(\nabla_{R^{(0)}} E_{\mathrm{joint}} + \nabla_{R^{(1)}} E_{\mathrm{joint}}\right) \\ &= \frac{F^{(0)} + F^{(1)}}{2} = \widehat{F}(R;t). \end{aligned} \tag{17}$$

Computing the two partial gradients and then averaging them implements this chain rule without requiring a differentiation path through coordinate preprocessing.

E.2 JOINT CENTERING

Suppose preprocessing subtracts a common center $c(R,t) \in \mathbb{R}^3$ from all $2N$ coordinates:

$$\widetilde{R}^{(0)} = R - \mathbf{1}_N c^\top, \qquad \widetilde{R}^{(1)} = R + \mathbf{1}_N t^\top - \mathbf{1}_N c^\top. \tag{18}$$

The argument does not require a particular choice of center. It requires the joint energy to be invariant to a common translation of both coordinate arrays, as for a model based on relative geometry. Write $A_o = \nabla_{\widetilde{R}^{(0)}} E_{\text{joint}}$ and $A_d = \nabla_{\widetilde{R}^{(1)}} E_{\text{joint}}$. Differentiating common-translation invariance gives

$$\mathbf{1}_N^\top (A_o + A_d) = 0. \tag{19}$$

At fixed t, the differential of the centered molecular energy is therefore

$$\begin{aligned} \mathrm{d}\widehat{E} &= \frac{1}{2}\langle A_o, \mathrm{d}R - \mathbf{1}_N \mathrm{d}c^\top \rangle_{\mathrm{F}} + \frac{1}{2}\langle A_d, \mathrm{d}R - \mathbf{1}_N \mathrm{d}c^\top \rangle_{\mathrm{F}} \\ &= \frac{1}{2}\langle A_o + A_d, \mathrm{d}R \rangle_{\mathrm{F}}. \end{aligned} \tag{20}$$

Here $\langle \cdot, \cdot \rangle_{\mathrm{F}}$ is the Frobenius inner product. The contribution through c vanishes by Equation (19), so averaging the copy forces computed on the centered graph gives the same molecular-coordinate gradient. This statement relies on common-translation invariance and does not establish invariance to changes in the relative offset.

E.3 FORCE-LOSS DECOMPOSITION AND MINIBATCH REDUCTION

Write $\mathcal{L}^{(0)} = \|F^{(0)} - F^*\|_{\mathrm{m}}^2$, $\mathcal{L}^{(1)} = \|F^{(1)} - F^*\|_{\mathrm{m}}^2$, and $\mathcal{L}_{\text{cons}} = \|F^{(0)} - F^{(1)}\|_{\mathrm{m}}^2$, where $\|u\|_{\mathrm{m}}^2 = (3N)^{-1} \sum_{i=1}^N \|u_i\|_2^2$ denotes the mean squared Cartesian component of a collection of per-atom vectors $u = (u_1, \ldots, u_N)$. The force term of Equation (9) is $\mathcal{L}_F = (\mathcal{L}^{(0)} + \mathcal{L}^{(1)} + \lambda \mathcal{L}_{\text{cons}})/(2+\lambda)$. Let $\widehat{F} = (F^{(0)} + F^{(1)})/2$ and $F_- = (F^{(0)} - F^{(1)})/2$. Since $F^{(0)} = \widehat{F} + F_-$ and $F^{(1)} = \widehat{F} - F_-$, expansion gives

$$\begin{aligned} \mathcal{L}^{(0)} + \mathcal{L}^{(1)} &= \|\widehat{F} - F^* + F_-\|_{\mathrm{m}}^2 + \|\widehat{F} - F^* - F_-\|_{\mathrm{m}}^2 \\ &= 2\|\widehat{F} - F^*\|_{\mathrm{m}}^2 + 2\|F_-\|_{\mathrm{m}}^2, \\ \mathcal{L}_{\text{cons}} &= \|F^{(0)} - F^{(1)}\|_{\mathrm{m}}^2 = 4\|F_-\|_{\mathrm{m}}^2. \end{aligned} \tag{21}$$

Substituting these expansions gives

$$\mathcal{L}_F = \frac{2}{2+\lambda}\|\widehat{F} - F^*\|_{\mathrm{m}}^2 + \frac{2(1+2\lambda)}{2+\lambda}\|F_-\|_{\mathrm{m}}^2. \tag{22}$$

At $\lambda = 0$, separate supervision of both copies still penalizes disagreement. Changing λ in this normalized objective also changes the coefficient on averaged-force fitting relative to a fixed energy-loss weight.

For a minibatch $\mathcal{B}$ of B configurations, let N_s be the original atom count, q_s the configuration weight, and $q_{E,s}, q_{F,s}$ the energy and force property weights. The code implements

$$\mathcal{L}_{E,\mathcal{B}} = \frac{1}{B}\sum_{s\in\mathcal{B}} q_s q_{E,s} \left(\frac{\widehat{E}_s - E_s^*}{N_s}\right)^2, \qquad \|A\|_{\mathrm{m},\mathcal{B}}^2 = \frac{\sum_{s\in\mathcal{B}} q_s q_{F,s} \sum_{i=1}^{N_s} \|A_{s,i}\|_2^2}{3\sum_{s\in\mathcal{B}} N_s}. \tag{23}$$

The force terms use this common weighted component mean, rather than a mean normalized by the sum of data weights. Equal-size, unit-weight configurations recover the average of the single-configuration objectives in Section 4.2. With repeated force labels on both copies, the ordinary all-$2N$-node force MSE equals $(\mathcal{L}^{(0)} + \mathcal{L}^{(1)})/2$; multiplying it by two gives the two supervised copy terms in Equation (9). The implementation applies the same linear weighted reduction to the original, duplicate, and consistency terms, so the mixed terms cancel and the decomposition remains valid.

E.4 Relative duplicate coordinates

Introduce coordinate arrays $u = (R^{(0)} + R^{(1)})/2$ and $v = R^{(1)} - R^{(0)}$, with $u, v \in \mathbb{R}^{N\times 3}$, and define

$$\mathcal{U}_\theta(u, v) = \frac{1}{2} E_{\text{joint}}(u - v/2, u + v/2). \tag{24}$$

Each u_i moves a corresponding atom pair together, whereas v_i changes the relative position within that pair. Applying the chain rule yields

$$\begin{aligned} -\nabla_u \mathcal{U}_\theta &= -\frac{1}{2}\left(\nabla_{R^{(0)}} E_{\text{joint}} + \nabla_{R^{(1)}} E_{\text{joint}}\right) = \widehat{F}, \\ \nabla_v \mathcal{U}_\theta &= \frac{1}{4}\left(-\nabla_{R^{(0)}} E_{\text{joint}} + \nabla_{R^{(1)}} E_{\text{joint}}\right) = \frac{F^{(0)} - F^{(1)}}{4} = \frac{F_-}{2}. \end{aligned} \tag{25}$$

Thus $\|F_-\|_{\text{m}}^2 = 4\|\nabla_v \mathcal{U}_\theta\|_{\text{m}}^2$. The disagreement term can be interpreted as a penalty on first-order sensitivity to relative duplicate coordinates. Training samples have $v_i = t$ for all i, although the partial derivative with respect to v has one three-dimensional component per atom pair. Penalizing these derivatives at sampled configurations does not establish invariance under arbitrary finite duplicate deformations. The decomposition identifies what the objective penalizes; it does not by itself determine whether that penalty improves generalization.

E.5 Fixed multi-displacement readout

For a fixed collection of K translations, energy and force predictions are averaged before error metrics are computed:

$$\widehat{E}_K(R) = \frac{1}{K}\sum_{k=1}^{K} \widehat{E}(R; t_k), \qquad \widehat{F}_K(R) = \frac{1}{K}\sum_{k=1}^{K} \widehat{F}(R; t_k). \tag{26}$$

Because the offsets do not vary with R during differentiation,

$$-\nabla_R \widehat{E}_K(R) = \frac{1}{K}\sum_{k=1}^{K} \left[-\nabla_R \widehat{E}(R; t_k)\right] = \widehat{F}_K(R). \tag{27}$$

This relation is conditional on using the same fixed offsets for energy and force. Resampling offsets during a trajectory changes the energy function being evaluated and is not covered by a claim of dynamics under one fixed potential. Multi-offset averaging evaluates multiple joint graphs; it is distinct from averaging copies already computed within one graph.