PhAST: Physics-Aware, Scalable, and Task-specific GNNs for accelerated catalyst design

Anonymous Author(s)
Affiliation
Address
email

Abstract

Mitigating the climate crisis requires a rapid transition towards lower carbon energy. Catalyst materials play a crucial role in the electrochemical reactions involved in this transition, such as electrofuel synthesis, renewable fertiliser production and energy storage. In this context, there is a need to discover more effective catalysts for these reactions. Machine learning (ML) holds the potential to efficiently model the properties of materials from large amounts of data, and thus to accelerate electrocatalyst design. The Open Catalyst Project OC20 data set was constructed to that end. However, most existing ML models trained on it are still neither scalable nor accurate enough for practical applications. Here, we propose several task-specific innovations, applicable to most architectures, which increase both computational efficiency and precision. In particular, we aim to improve (1) the graph creation step, (2) atom representations and (3) the energy prediction head. We describe and evaluate these contributions across several architectures, showing up to 5$\times$ reduction in inference time without sacrificing accuracy.

1 Introduction

In order to mitigate climate change at a global scale, it is imperative to develop scalable infrastructure for storing low-carbon energy at scale and utilising it in industrial chemical processes. Such processes rely on electrochemical reactions, and scaling them requires the design of more efficient, low-cost catalysts [21]. However, catalyst discovery is held back by the high compute requirements of quantum mechanical simulations like Density Functional Theory (DFT), which constrains researchers to testing only a limited number of candidate materials. Machine learning (ML) holds the potential to approximate these calculations while reducing the time needed to assess each candidate by several orders of magnitude [22].

To enable to use of ML for catalyst discovery, the Open Catalyst Project released OC20 [3], a large data set of adsorbate-catalyst systems and their *relaxed* energy—a relevant metric for the efficiency of a catalyst—computed with DFT from the initial atomic structure. Despite recent progress [7, 20], major challenges remain. First, state-of-the-art models have not yet reached high enough performance for practical applications. Second, they are too computationally expensive to allow the millions of inferences required to explore the large space of potential catalysts. Third, the graph neural networks (GNN) typically used were designed for general 3D molecular prediction tasks rather than specifically for catalyst discovery, the complexity of which may benefit from task-specific architectures.

To address these challenges, we propose multiple model improvements to increase the accuracy and scalability of generic GNNs applied to catalyst discovery. In particular, our contributions are: (1) a graph representation that is tailored to the task at hand, (2) richer physics-based atom representations, and (3) an energy head that learns a weighted average of per-atom predictions. We provide a thorough evaluation of these contributions on OC20 and a full ablation study, improving performance by 5-8%

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

while dividing compute time by a factor of 3 to 5. We believe our work provides valuable insights for future research as it leverages domain-specific knowledge to improve parts of the pipeline that were not investigated up to now.

2 Background

The problem we address is the prediction of the relaxed energy $y \in \mathbb{R}$ of an adsorbate-catalyst system from its initial configuration in space $(\mathbf{X}, \mathbf{Z})$, where $\mathbf{X} \in \mathbb{R}^{N \times 3}$ is the matrix of 3D atom positions and $\mathbf{Z} \in \mathbb{N}^N$ contains atom characteristic numbers. This is commonly represented as a graph regression task, where each sample is represented as a 3D graph $\mathcal{G}$ with node set $\mathcal{V}$ of dimension N and adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. $\mathbf{H} \in \mathbb{R}^{N \times H}$ represents atom embeddings and $\mathbf{T} \in \{0, 1, 2\}^N$ corresponds to tag information (see 3.1). A variety of ML models have been applied to this task, all of which adopted a Graph Neural Network architecture as it naturally fits 3D molecular prediction. All such GNNs share a common pipeline for how they are applied, as depicted in Fig. 1.

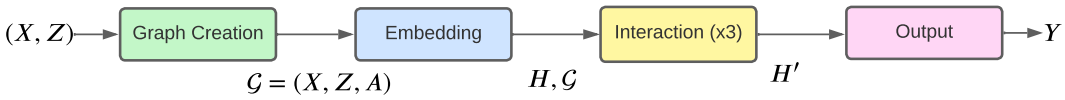

Figure 1: Common GNN inference pipeline for 3D molecular prediction. The graph creation step remains unchanged across all methods: it creates $\mathbf{A}$ using cutoff distances and periodic boundary conditions. The Embedding and Output blocks slightly differ but the core underlying idea is the same. The Embedding block learns a representation for each chemical element and the Output applies a global pooling of each node's representation to obtain the energy prediction. The key distinction between methods lies essentially in the Interaction block, where the message passing schemes vary.

In 3D molecular prediction tasks, it is desirable to endow ML models with relevant symmetry properties. In particular, we want predictions to be invariant to translations, rotations and (often) reflections. Many models enforce these physical priors within the architecture, making it explicitly invariant or equivariant to the desired transformations. Formal definitions are included in Appendix A.1.

Many GNNs in prior work focus on enforcing equivariance, though it is not strictly required for relaxed energy prediction, which calls for invariance. Besides, equivariant GNNs [16, 6, 1, 2] are very expensive computationally as they are constrained by equivariant filters built on spherical harmonics and the Clebsch-Gordan [2] tensor product. Alternatively, E(3)-invariant methods do not use atom positions directly in their internal workings. Instead, these methods extract and use quantities that remain invariant under rotations and reflections. Dimenet++ [11, 10], for example, offers a directional message passing (MP) mechanism that incorporates bond angles in addition to atom relative distances. However, distances and bond angles do not suffice to uniquely identify the graph 3D structure. This is achieved by SphereNet [12] and GemNet [7], which additionally extracts torsion information (between quadruplets of nodes). On the downside, these methods are very computationally expensive as they require considering 3-hop neighbourhoods for each update step. Importantly, all these MP methods aim at broad applicability and do not leverage the specific constraints of individual tasks.

3 Proposed Method

In this section, we describe PhAST, our proposed framework with architectural innovations, which is applicable to most current GNNs used in catalyst discovery. These include a new graph creation step, richer atom representations and an advanced energy head for graph-level prediction.

3.1 Graph creation

Although the graph construction step is critical in graph ML tasks, previous work on the OC20 data set has paid little or no attention to it. Most methods reuse the original proposal in [3]. In particular, each graph's atoms are tagged as part of the adsorbate (tag 2), the catalyst's surface (tag 1), or its sub-surface volume (tag 0). Tag 0 atoms were originally added to represent more explicitly the repeating pattern of the slab. They are, by definition, further away from the adsorbate and are fixed by construction, unlike tag 1 and tag 2 atoms, which can move during the relaxation. As a result, we

hypothesise they contain redundant information, making them of lesser importance to predict the final relaxed energy. Besides, since they account for $\sim 65\%$ of the nodes B.3, and since SOTA GNNs often depend on multiple hops to compute bond and torsion angles, we propose to remove these nodes from the graph. This should greatly reduce inference time without impacting expressivity. Additionally, we explore forming *super nodes* that aggregate tag 0 atoms to avoid a potential information loss caused by their total removal. We briefly describe these changes below, with more details in A.2.

remove-tag-0 removes all atoms with tag 0 ($t_i = 0$) from the graph, adapting correspondingly all graph attributes ($\mathbf{X}$, $\mathbf{A}$, $\mathbf{Z}$, $\mathbf{T}$, etc.).

one-supernode-per-graph aggregates all tag 0 nodes from $\mathcal{G}$ into a supernode s with position $\mathbf{x}_s = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \mathbf{x}_i$, adjacency $A_{is} = \max(A_{ij} : j \in \mathcal{S})$ and a new z_s.

one-supernode-per-atom-type replicates the above strategy but creates one super node per chemical element in the catalyst subsurface. Its attributes are defined based on its components, as previously.

3.2 Atom Embeddings

In all previous GNNs, atom representations are learned from scratch based on atomic number $\mathbf{H} = \mathbf{H_Z}$. We propose to leverage domain information to improve these representations. **First**, we hypothesise that whether a given atom belongs to the adsorbate, the catalyst surface, or its subsurface is important information. We therefore include tag information as an additional embedding $\mathbf{H_T}$. **Second**, we know from previous studies that some atomic properties (e.g. atomic radius or density) are useful for catalyst discovery [15, 18]. We leverage them as an additional embedding vector $\mathbf{H_F}$ (see A.3 for the full list of properties). **Lastly**, we include an embedding for the group and period information ($\mathbf{H_P}, \mathbf{H_G}$) since atoms belonging to the same group or period often share similar behaviours [19]. As a result, our proposed atom embedding $\mathbf{H}$ is a concatenation of all of the above.

3.3 Energy head

In the literature, there is often limited focus on the energy head, which is the part of the output block responsible for the energy computation from final atom representations $\mathbf{h}_i^L$. All GNNs compute the relaxed energy using global pooling $y = \sum_{i\in\mathcal{V}} h_i$, where node embeddings are reduced to a scalar h_i by linear layers. We identify two limitations in this procedure: First, all atoms are assigned the same importance, even though the properties of an atom are normally influenced by what element it is. Second, the graph topology is neglected by simply summing all atom encodings regardless of their 3D positions. To overcome these limitations, we explore new energy heads.

First, **a weighted average of node representations**, which grants adaptive importance to each chemical element, expressed as $\hat{y} = \sum_{i\in\mathcal{V}} \alpha(\mathbf{h}_i^L) \cdot h_i$ or $\hat{y} = \sum_{i\in\mathcal{V}} \alpha(\mathbf{h}_i^0) \cdot h_i$, where the learnable importance weights $\alpha(\cdot)$ depend either on the embedding block initial encodings $\mathbf{h}_i^0$ or final ones $\mathbf{h}_i^L$.

Second, **a hierarchical pooling approach** endowed with the following energy head pipeline: $\mathbf{h}_i^L \rightarrow$ [Pooling $\rightarrow$ GCN] ($\times 2$) $\rightarrow$ Global Pooling $\rightarrow$ MLP $\rightarrow$ $\hat{y}$. By applying a graph convolutional network (GCN) [9] on a coarsened graph, we propagate information differently, allowing us to capture hierarchical graph information. We implement two pooling operators: (1) GRACLUS [4]: a deterministic way to regroup topologically close nodes together based on the mincut problem. (2) HOSCPOOL [5]: an end-to-end operator that learns a cluster assignment matrix using a loss function inspired by motif spectral clustering. Contrary to GRACLUS, it leverages node features, captures higher-order connectivity patterns and is differentiable.

4 Evaluation

In this section, we evaluate the performance and scalability of our contributions for three well-known GNNs on the OC20 dataset [3].

Dataset. OC20 contains 1,281,040 DFT relaxations of randomly selected catalysts and adsorbates from a set of plausible candidates. In this paper, we focus on the IS2RE task [22], that is the direct prediction of the relaxed adsorption energy from the initial atomic structure. It comes with a pre-defined train/val/test split, 450,000 training samples and hidden test labels. Experiments are evaluated on the validation set, which has four splits of $\sim 25K$ samples: In Domain (ID), Out of

Baseline / MAE	ID	OOD-ad	OOD-cat	OOD-both	Average	Inference time (s)
SchNet	0.637	0.734	0.661	0.703	0.683	15.04 ± 0.49
*SchNet**	0.618	0.677	0.611	0.616	**0.630**	$\mathbf{5.26 \pm 0.36}$
D++	0.571	0.722	0.561	0.661	0.628	110.11 ± 0.57
*D++**	0.568	0.654	0.560	0.597	**0.595**	$\mathbf{20.49 \pm 0.63}$
ForceNet	0.658	0.701	0.632	0.628	0.654	167.08 ± 0.52
*ForceNet**	0.612	0.664	0.592	0.597	**0.616**	$\mathbf{33.18 \pm 0.60}$

Table 1: Comparing model performances on *OC20* IS2RE. *Average* is computed over all validation splits. The $*$ refers to our PhAST baselines, which all show improved performance and drastic speedups. Note that SchNet* almost matches the original DimeNet++ while being 21 times faster.

Domain Adsorbates (OOD-ads), Out of Domain catalysts (OOD-cat), and Out of Domain Adsorbates and catalysts (OOD-both). Performance is measured by the energy Mean Average Error (MAE) on each val splits. Scalability is assessed through inference time over the ID validation set (in seconds).

Baselines. We study the enhancements brought by our components to three key GNNs for molecular predictions: SchNet [14], DimeNet++ [10] and ForceNet [8]. We describe their main details in Appendix B.1. We use the hyperparameters, training settings and model architectures provided in the original papers. We compare every baseline with their PhAST counterpart, incorporating the best components of each category, that is graph creation (3.1), enriched atom embedding (3.2) and advanced energy-head (3.3), as determined by an ablation study (see below and Appendix B.2).

Ablation. Sub-surface atoms appear to contain redundant information as *remove-tag-0* does not cause performance drop and aggregating it into super nodes does not yield better results. Besides, it dramatically decreases compute time. Enriched atom embeddings improve performance and generalisation, especially when adding tag information. Hierarchical pooling approaches are not very successful, either due to the difficulty of the task or the absence of hierarchical structures, but both energy-head weighted averages are beneficial. For a finer understanding of individual contributions, refer to the full ablation study in Appendix B.2. In conclusion, we obtain the following best components for the PhAST version of our models: remove-tag-0, full concatenation of atom embeddings and predicting the system energy as a learned weighted average of per-atom predictions.

Results. From Table 1, we conclude that our set of PhAST enhancements consistently improve both performance and inference time upon the original baselines. More precisely, we improve average MAE over the four validation splits by 7.75%, 5.25% and 5.80% compared to SchNet, D++ and ForceNet, while reducing model inference time by factors of 2.85, 5.25, and 5.03, respectively. Moreover, we observe an MAE improvement of 12.3% over SchNet and 9.0% over DimeNet++ on val-both-ood (compared to a 7.75% and 5.25% decrease in mean MAE). This points to the fact that PhAST models generalise better than the original baselines. From the ablation study conducted in B.2, this is due to the combination of our extensions, as they all contribute to significantly better performance on out-of-distribution adsorbate-catalyst systems (OOD-both). Finally, note that inference time gains are doubled from SchNet, a 1-hop message passing (MP) approach, to DimeNet++, a 2-hop MP approach (from 2.85 to 5.25). We can therefore expect even better computational scaling gains on newer models like GemNet[7], which builds on 3-hop MP methods.

5 Conclusion

In this work, we presented several enhancements targeted to catalyst discovery and applicable across existing GNN models. We showed that (1) enriching atom representations with physics-based properties, (2) weighting atoms' importance when computing the system energy, and (3) tailoring the graph creation to the specific task at hand reduce inference time by a factor of 3-5 with better accuracy. Our results suggest that complex practical applications like catalyst discovery benefit from task-specific methods rather than general 3D molecular prediction GNNs.

References

[1] 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):1–11, 2022.

[2] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik Bekkers, and Max Welling. Geometric and physical quantities improve e (3) equivariant message passing. *arXiv preprint arXiv:2110.02905*, 2021.

[3] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, 2021.

[4] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.

[5] Alexandre Duval and Fragkiskos Malliaros. Higher-order clustering and pooling for graph neural networks. *arXiv preprint arXiv:2209.03473*, 2022.

[6] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

[7] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

[8] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C Lawrence Zitnick. Forcenet: A graph neural network for large-scale quantum calculations. *arXiv preprint arXiv:2103.01436*, 2021.

[9] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[10] Johannes Klicpera, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

[11] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[12] Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021.

[13] Łukasz Mentel. mendeleev – a python resource for properties of chemical elements, ions and isotopes.

[14] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[15] Ichigaku Takigawa, Ken-ichi Shimizu, Koji Tsuda, and Satoru Takakusagi. Machine-learning prediction of the d-band center for metals and bimetals. *RSC advances*, 6(58):52587–52595, 2016.

[16] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] Logan Ward, Ruoqian Liu, Amar Krishna, Vinay I Hegde, Ankit Agrawal, Alok Choudhary, and Chris Wolverton. Including crystal structure attributes in machine learning models of formation energies via voronoi tessellations. *Physical Review B*, 96(2):024104, 2017.

[19] Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018.

[20] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

[21] Behnam Zakeri and Sanna Syri. Electrical energy storage systems: A comparative life cycle cost analysis. *Renewable and sustainable energy reviews*, 42:569–596, 2015.

[22] C Lawrence Zitnick, Lowik Chanussot, Abhishek Das, Siddharth Goyal, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Thibaut Lavril, Aini Palizhati, Morgane Riviere, et al. An introduction to electrocatalyst design using machine learning for renewable energy storage. *arXiv preprint arXiv:2010.09435*, 2020.

A Method

A.1 Invariance and equivariance to symmetries

Let $\phi : V \to \mathbb{R}$ and $\Phi : V \to W$ be arbitraty functions where W,V are linear spaces. Let G be a group describing a symmetry which we want to incorporate into ϕ, Φ (e.g. euclidean symmetries $E(3)$). We use group representations $\rho_1 : G \to GL(V)$ and $\rho_2 : G \to GL(W)$, where $GL(V)$ is the space of invertible linear maps $V \to V$ to represent how the symmetries $g \in G$ are applied to vectors $X \in V, W$. ϕ is an invariant function if it satisfies $\phi(\rho_1(g)X) = \phi(X), \forall g \in G$ and $X \in V$. Φ is an equivariant function if it satisfies $\Phi(\rho_1(g)X) = \rho_2(g)\Phi(X), \forall g \in G$ and $X \in V$.

In this paper, we focus on accelerated catalysis and thus on adslab relaxed adsorption energy prediction. Like for most 3D molecular prediction tasks, we want GNNs to predict the same energy for two rotated, translated or reflected versions of the same system, since their energy is equal in real-life. Hence, we target $E(3)$-invariant models, where $E(3)$ is the Euclidean group in a 3D space (we have 3D atom positions), that is, the transformations of that 3D space that preserve the Euclidean distance between any two points (i.e. rotations, reflections, translations). Note that we do desire reflection invariance because we rotate the whole adsorbate-catalyst system and not just the adsorbate, in which case chiral molecules may have a different behaviour and shall be considered distinctly.

As implied by the above definitions, invariance is a specific case of equivariance, which is a more general and powerful property. Although desired for several molecular property prediction tasks, equivariance is not perfectly adapted for accelerated catalysis where only invariance is required: the energy of the system does not change with those transformations. Since enforcing equivariant predictions is difficult and often computationnally expensive[1], we grant a smaller importance to equivariant GNNs in our paper.

A.2 Graph creation

A.2.1 OC20

Chanussot, Lowik, et al. [3] create each OC20 sample by choosing a bulk material from the Materials Project database[2]. Then, they select a surface from the bulk using Miller indices (at random) and

[1] Equivariant methods combine node features (learned based on atom positions) with a continuous equivariant filter, constructed based on spherical harmonics and learnable radial function, via a Clebsh-Gordan tensor product to guarantee an E(3) or SE(3) equivariant output.

[2] `https://materialsproject.org/`

replicate it at depth of at least 7 Å and a width of at least 8 Å. The final slab is defined by a unit cell that is periodic in all directions with a vacuum layer of at least 20 Å applied in the z direction. Next, they pick a binding site on this surface to attach the adsorbate onto the catalyst. The graph is now a set of atoms with their 3D positions. Last but not least, edges are created between any two nodes within a cutoff distance $c = 6$Å of each other (considering periodic boundary conditions).

A.2.2 PhAST graph creation process

Although well grounded, the assumptions of this graph creation process are rarely questioned. We do, with the objective of making the graph sparser and more informative for subsequent GNNs. We describe more formally the three proposals evoked in 3.1.

remove-tag-0. We denote the set of tag 0 atoms by $\mathcal{S} = \{i \in \mathcal{V} : t_i = 0\}$. The graph we create has attributes $\mathbf{X} = \mathbf{X}_{\mathcal{S}}$ where $\mathbf{X}_{\mathcal{S}}$ is the position of all atoms except those in $\mathcal{S}$. Similarly for $\mathbf{Z} = \mathbf{Z}_{\mathcal{S}}$ and $\mathbf{T} = \mathbf{T}_{\mathcal{S}}$. $\mathbf{A}$ is defined as usual based on cutoff distance (and pbc): $A_{ij} = 1$ if $||\mathbf{x}_i - \mathbf{x}_j|| < c$, 0 otherwise. The remaining graph attributes (cell offsets, distances, etc.) are updated correspondingly.

one-supernode-per-graph. The position of the new super node is the mean of its components: $\mathbf{x}_s = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \mathbf{x}_i$ (with $\mathcal{S}$ as defined above), and we associate it to a new characteristic number z_s (corresponding to a new element in atomic table) and adjacency $A_{is} = \max(a_{ij} : j \in \mathcal{S})$. We now remove all tag-0 atoms using the above method, and finally add a tag-0 attribute $t_i = 0$ to the supernode. Note that we also remove self-loop for the supernode.

one-supernode-per-atom-type. This extension is similar to the previous one, except that we create one supernode for each chemical element in the sub-surface catalyst. This complexify a bit the graph definition. Let a be the number of distinct elements with tag 0 in the graph ($a = 1, 2, 3$ by construction) and a_k be their characteristic number. Let $\mathcal{S}_{a_k} = \{i \in \mathcal{V} | t_i = 0 \text{ and } z_i = a_k\}$ be the set of atoms corresponding to each distinct tag-0 element a_k. Each supernode s_k is defined with $x_{s_k} = \frac{1}{|\mathcal{S}_{a_k}|}\sum_{i\in\mathcal{S}_{a_k}} \mathbf{x}_i$, $z_k = a_k$, $A_{is_k} = \max(A_{ij} : j \in \mathcal{S}_{a_k})$ ($\forall i \in \mathcal{V}$) and $A_{s_{k'}s_k} = 1$, $A_{s_k s_k} = 0$.

For both super-node methods, we encode the number of tag-0 nodes aggregated into each super node with Positional Encodings ([17]) to represent their "cardinal".

A.3 Atom properties for the Embedding block

In atom embeddings, we use the following properties from the `mendeleev` Python package ([13]):

1. atomic radius,
2. atomic volume
3. atomic density
4. dipole polarizability
5. electron affinity
6. electronegativity (allen)
7. Van-Der-Walls radius
8. metallic radius
9. covalent radius
10. ionization energy (first and second order).

B Experiments and ablation study

B.1 Baseline description and hyperparameters

We target three well-known GNN baselines to study the impact of our contributions. As evoked in Fig 1 of the paper, they all follow a similar pipeline, mainly differing in their interaction blocks, which we briefly detail below.

SchNet [14] is a simple message passing architecture that leverages relative distances to update atom representations via a continuous filter: $\mathbf{h}_i^{(l+1)} = \sum_j \mathbf{h}_j^l \odot W^l(\mathbf{x}_i - \mathbf{x}_j)$ where $W^l(\mathbf{x}_i - \mathbf{x}_j)$ is a radial basis function to encode distance between atom pairs.

DimeNet++ [10] is an optimised version of DimeNet [10], which proposes a directional message passing. In different terms, they compute and update edge representations instead of atoms) using interatomic distances $\mathbf{e}_{RBF}$ (encoded via bessel functions) and bond angles $\mathbf{a}_{SBF}$ (encoded via 2D spherical Fourier-Bessel basis):

$$\mathbf{m}_{ij}^{(l+1)} = f_{update}\big(\mathbf{m}_{ij}^{(l)}, \sum_{k \in N_j \setminus i} f_{int}(\mathbf{m}_{ij}^{(l)}, \mathbf{e}_{RBF}^{(ij)}, \mathbf{a}_{SBF}^{(ki,ji)})\big)$$

ForceNet [8] is a scalable force-centric GNN that does not impose explicit physical constraint (energy conservation, rotational invariance), thus avoiding some memory intensive computations. It still attempts to enforce invariance by efficient rotation-based data augmentation. Model-wise, it adopts a node message passing approach that leverages node positions directly via a spherical harmonics basis.

Choice of baselines. In this paragraph, we briefly justify why we focus on these three baselines. SchNet and DimeNet++ are probably the two most famous models for OC20. They were already implemented and trained for the OC20 IS2RE task, and showcase a different granularity of information (D++ has bond angles instead of atomic distances alone). They are evident choices to test the relevance of our contributions. Then, we chose ForceNet because it uses atom positions directly in its message passing inner workings and because it is the only method that does not enforce symmetries explicitly in its architecture. We decided to limit ourselves to these three methods, mainly because we believe that they are representative enough for this use-case. Besides, we did not want to spend too much computational resources testing these contributions on more models. On this note, we avoided equivariant GNNs because of their computational complexity and because we believe they are not perfectly adapted for catalyst discovery. Similarly, we avoided the SOTA invariant method GemNet [7] because of how expensive it is to train. Moreover, it remains quite similar to D++ in its inner workings and architectural design.

Hyperparameters. We use each method's optimal set of parameters, provided in the config folder of the OCP repository for the IS2RE task, for the full dataset: `https://github.com/Open-Catalyst-Project/ocp/tree/main/configs/is2re/all`. Since ForceNet was not applied to IS2RE before, we adapted its S2EF configuration file to fit the IS2RE task. The only change is the smaller number of epochs used for DimeNet++ (10 instead of 20) and SchNet (20 instead of 30), as these additional epochs only lead to a small performance gain for a large amount of additional compute time.

B.2 Ablation study

Notation. For the embedding block, *tag-embed* defines atom embeddings $\mathbf{H}$ using atom tag information and characteristic number: $\mathbf{H} = \mathbf{H}_Z || \mathbf{H}_T$, where $||$ denotes concatenation. Similarly, *phys-embed* defines $\mathbf{H} = \mathbf{H}_Z || \mathbf{H}_F$. *l-phys-embed* is a learnable alternative $\mathbf{H} = \mathbf{H}_Z || MLP(\mathbf{H}_F)$. *pg* refer to period and group embeddings: $\mathbf{H} = \mathbf{H}_Z || \mathbf{H}_P || \mathbf{H}_G$. *All* is a concatenation of all five embeddings: $\mathbf{H} = \mathbf{H}_Z || \mathbf{H}_F || \mathbf{H}_G || \mathbf{H}_P || \mathbf{H}_T$. For the graph creation step, we have defined these approaches in A.2.2 (note: *sn* stands for supernode). For the energy head part, *w-init* (*w-final*) denote the weighted average of initial (final) atom embeddings. *graclus* and *hoscpool* refer to the two hierarchical pooling approaches.

See Tables 2, 3, 4. The results are reported similarly to Table 1.

B.3 Graph-Rewiring: impact on the number of edges and nodes

Table 2: SchNet ablation study on *OC20* IS2RE.

Method / MAE	Average	ID	OOD-ad	OOD-cat	OOD-both	Inference time (s)
tag-embed	**0.648***	**0.637**	**0.690***	**0.629***	0.638*	15.390 +/- 0.337
phys-embed	0.662*	0.644	0.700*	0.639*	0.654*	15.482 +/- 0.389
l-phys-embed	0.678*	0.650	0.733*	0.649*	0.679*	15.590 +/- 0.377
pg	0.673*	0.646	0.725*	0.644*	0.676*	15.441 +/- 0.528
All	0.659*	0.665	**0.690***	0.651*	**0.630***	15.469 +/- 0.461
remove-tag-0	**0.648***	**0.627***	**0.705***	0.627*	**0.634***	**4.749 +/- 0.501**
sn-graph	0.654*	0.633*	**0.705***	0.633*	0.646*	5.545 +/- 0.681
sn-atom-type	0.663*	0.628*	0.738	**0.626***	0.659*	6.386 +/- 0.453
w-init	**0.657***	**0.635***	0.715*	**0.631***	**0.646***	15.370 +/- 0.480
w-final	0.668*	0.647	**0.713***	0.644*	0.670*	15.418 +/- 0.385
graclus	0.950	0.910	0.994	0.946	0.950	16.655 +/- 0.605
hoscpool	0.667*	0.650	0.719*	0.636*	0.662*	53.931 +/- 1.369
SchNet	0.683	0.637	0.734	0.661	0.703	15.401 +/- 0.498

Table 3: Dimenet++ ablation study on *OC20* IS2RE*

Method / MAE	Average	ID	OOD-ad	OOD-cat	OOD-both	Inference time (s)
tag-embed	**0.579***	0.551*	**0.659***	0.545*	**0.594***	110.042 +/- 0.890
phys-embed	0.590*	0.561*	0.671*	0.555	0.606*	110.082 +/- 0.747
l-phys-embed	0.612*	0.566*	0.700*	0.557*	0.626*	110.121 +/- 0.755
pg	0.624*	0.564*	0.710*	0.568	0.652*	110.184 +/- 0.717
All	0.602*	**0.550***	0.691*	**0.540***	0.626*	110.034 +/- 0.770
remove-tag-0	0.610*	0.576	0.684*	0.568	0.627*	**20.152 +/- 0.553***
sn-graph	-	-	-	-	-	25.046 +/- 1.548
sn-atom-type	-	-	-	-	-	27.220 +/- 0.940
w-init	0.611*	0.568*	0.686*	**0.560***	0.630*	110.238 +/- 0.815
w-final	**0.601***	0.571	0.660*	0.566	**0.606***	110.169 +/- 0.769
graclus	0.620*	0.567*	0.701*	0.567	0.648*	-
hoscpool	0 .618*	**0.565***	0.703*	0.563	0.642*	252.559 +/- 0.455
D++	0.628	0.571	0.722	0.561	0.661	110.113 +/- 0.578

* Unfortunately we did not manage to train DimeNet++ with the super node extensions.
For a very wide range of learning rates, the training loss consistently reached NaN values.

Table 4: ForceNet ablation study on *OC20* IS2RE.

Method / MAE	Average	ID	OOD-ad	OOD-cat	OOD-both	Inference time (s)
tag-embed	0.640*	0.639*	0.690*	0.616*	0.617*	168.646 +/- 0.733
phys-embed	0.653*	0.657*	0.702	0.626*	0.627*	172.049 +/- 0.983
l-phys-embed	0.667	0.654*	0.734	0.626*	0.650	167.981 +/- 0.714
pg	**0.634***	0.644*	**0.669***	0.618*	**0.603***	170.701 +/- 0.966
All	0.637*	**0.622***	0.680*	**0.603**	0.615*	169.236 +/- 0.846
remove-tag-0	**0.628***	**0.637***	**0.668***	**0.611***	**0.598**	**17.065 +/- 0.584**
sn-graph	0.635*	0.640*	0.676*	0.617*	0.607*	55.309 +/- 1.861
sn-atom-type	0.632*	0.641*	0.672*	0.616*	0.601*	70.557 +/- 0.157
w-init	**0.639***	0.639	0.687*	**0.611***	0.666	170.719 +/- 0.600
w-final	0.660	0.655*	0.716	0.627*	0.644	170.720 +/- 1.618
graclus	0.671	0.622*	0.722	0.634	0.646	172.494 +/- 1.101
hoscpool	0.655	**0.621***	0.703	0.638	**0.638**	202.034 +/- 1.231
ForceNet	0.654	0.658	0.701	0.632	0.628	167.089 +/- 0.525

Rewiring	Atoms	Edges
Train		
Full graph	35 789 459	1 309 308 840
remove-tag-0	32.53%	16.61%
one-supernode-per-graph	33.81%	17.76%
one-supernode-per-atom-type	35.45%	19.09%
ID		
Full graph	1 939 553	70 825 106
remove-tag-0	32.54%	16.65%
one-supernode-per-graph	33.83%	17.80%
one-supernode-per-atom-type	35.47%	19.13%
OOD-ads		
Full graph	1 918 704	69 877 652
remove-tag-0	32.42%	16.50%
one-supernode-per-graph	33.72%	17.64%
one-supernode-per-atom-type	35.38%	18.97%
OOD-cat		
Full graph	1 917 954	70 314 085
remove-tag-0	32.88%	16.78%
one-supernode-per-graph	34.18%	17.95%
one-supernode-per-atom-type	35.90%	19.34%
OOD-both		
Full graph	2 094 709	80 074 123
remove-tag-0	31.12%	15.33%
one-supernode-per-graph	32.31%	16.37%
one-supernode-per-atom-type	34.17%	17.83%

Table 5: Comparison of the number of nodes and edges in the original 5 datasets (training and 4 validation splits) and the remaining number of nodes and edges after the various rewiring strategies are performed. We can see that our rewiring methods generally remove 65+% of the atoms and 80+% of the edges.