SUPPORTING INFORMATION
DASH: Dynamic Attention-Based Substructure Hierarchy for Partial Charge Assignment

Marc T. Lehner,[†a] Paul Katzberger,[†a], Niels Maeder,[a] Carl C. G. Schiebroek,[a] Jakob Teetz,[a] Gregory A. Landrum,[a] and Sereina Riniker[*a]

[a] *Department of Chemistry and Applied Biosciences, ETH Zürich, Vladimir-Prelog-Weg 2, 8093 Zürich, Switzerland. E-mail: sriniker@ethz.ch*

† *These authors contributed equally.*

Code Listings

Listing 1: Pseudocode to build a DASH as a tree structure

```
tree = Tree() # tree with root node
# df with every atom in the data set and a mol index
df = pd.read_csv("data/atom_features.csv")
for level in range(max_depth):
  df["max_attention"]=df.apply(get_neighbour_max_attention, axis=1)
  df = df.sort_values(by=["max_attention"], ascending=False)
  for line in df.itertuples():
    new_atom_feature = get_atom_feature(line)
    if tree.check_if_atom_feature_in_tree(new_atom_feature, level):
      node = get_node_with_atom_feature(new_atom_feature, level)
      node.update_statics(line)
    else:
      node = tree.add_node(new_atom_feature, level)
      node.update_statics(line)
```

Listing 2: Pseudocode to assign DASH partial charges to a molecule

```
def match_molecule(tree, mol, norm="std", symmetrize=True):
    # match all atoms separately in the tree
    charges = [match_atom_in_mol(tree, mol, atom) for atom in mol.GetAtoms()]
    # post processing
    charges = normalize_charges(charges, method=norm)
    if symmetrize:
        charges = symmetrize_charges(charges)
    return charges
```

Listing 3: Pseudocode to assign DASH partial charges to an atom

```
def match_atom_in_mol(tree, mol, atom):
    current_node = tree.root
    current_sub_graph = sub_graph(atom)
    total_attention = 0
```

```
for i in range(max_depth):
    possible_new_atom_features = get_sub_graph_neighbors(mol, current_sub_graph)
    found_match = False
        for current_node in current_correct_node.children:
            for possible_atom_feature in possible_new_atom_features):
                if possible_atom_feature in current_node.atoms:
                    current_correct_node = current_node
                    current_sub_graph.append(possible_atom_idx)
                    total_attention += current_node.attention
                    found_match = True
                    break
            if found_match:
                break
        if total_attention > attention_threshold:
            break
    return (current_correct_node.result)
```

Additional Figures

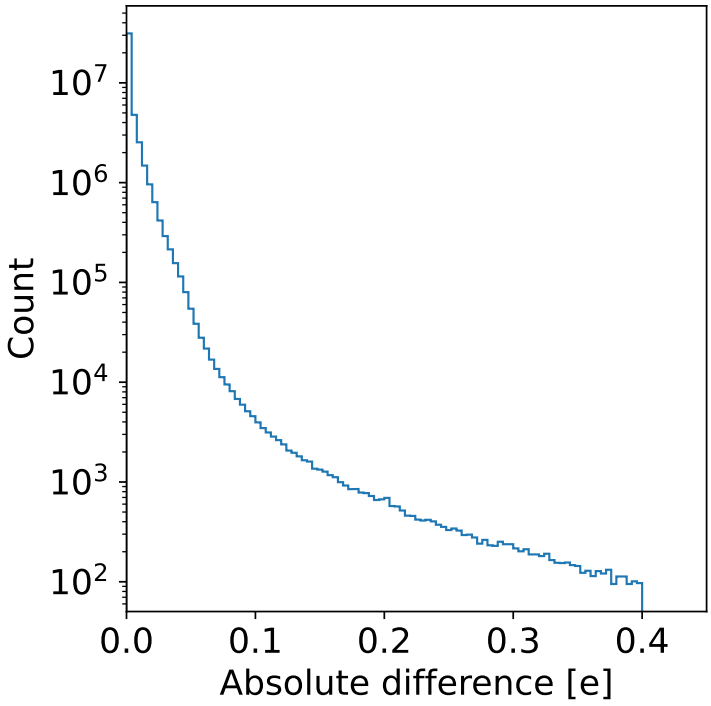

Figure S1: Histogram of absolute partial-charge differences for all molecules in the full dataset. For each atom, the difference between each conformer and the median of the three conformers (CNF) is shown.

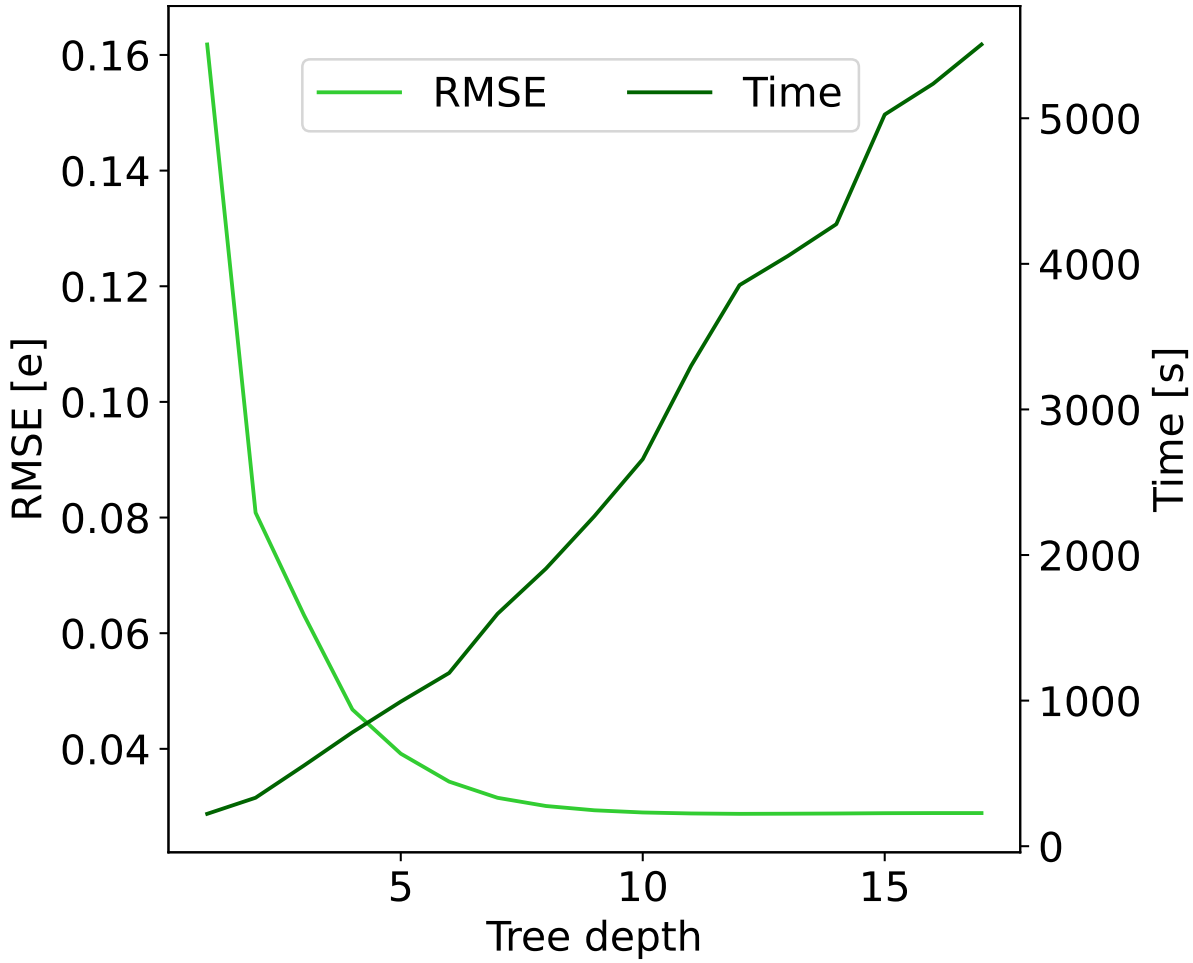

Figure S2: RMSE (light green, left axis) of the DASH partial charges with respect to the MBIS reference charges on the validation set and the time to match all atoms of all molecules in the validation set (dark green, right axis) as a function of the maximal depth. No attention threshold was used.

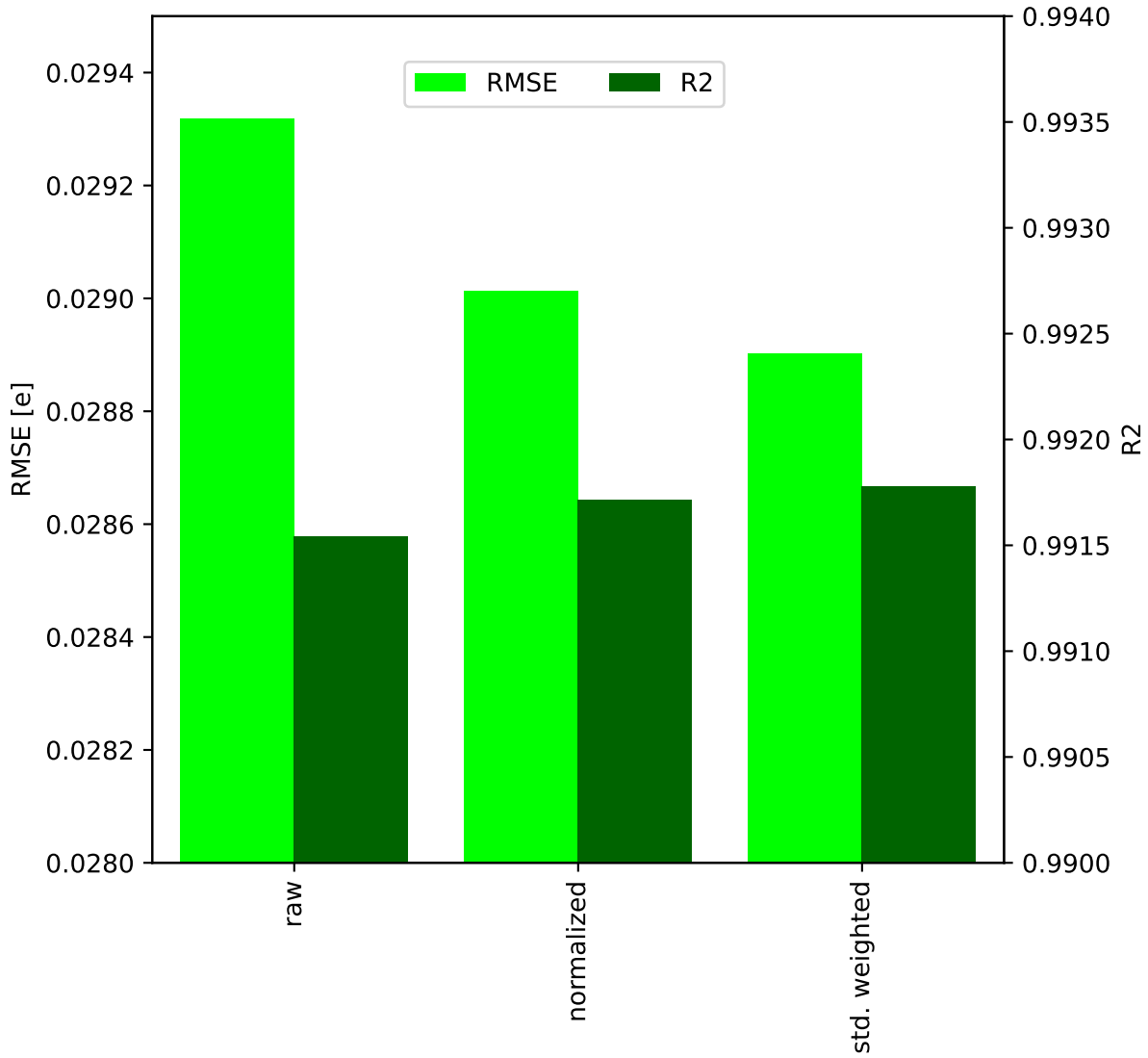

Figure S3: RMSE (light green) and R^2 (dark green) of DASH partial charges with respect to the MBIS reference charges on the validation set for the 'raw' charges, normalized with Eq. 3 in the main text, and normalized with Eq. 4 in the main text. A maximal depth of 16 layers and an attention threshold of 0.95 was used to construct the DASH tree structure.

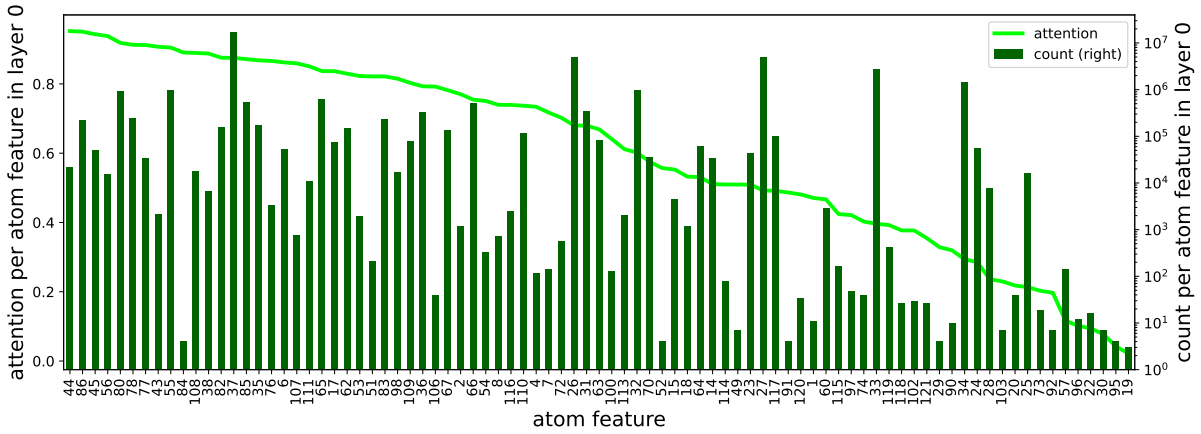

Figure S4: Attention value of the level 0 nodes in the DASH tree, which correspond to the 122 atom types, and the count (i.e., number of atoms with this type in the training set). The nodes are sorted by decreasing attention value.

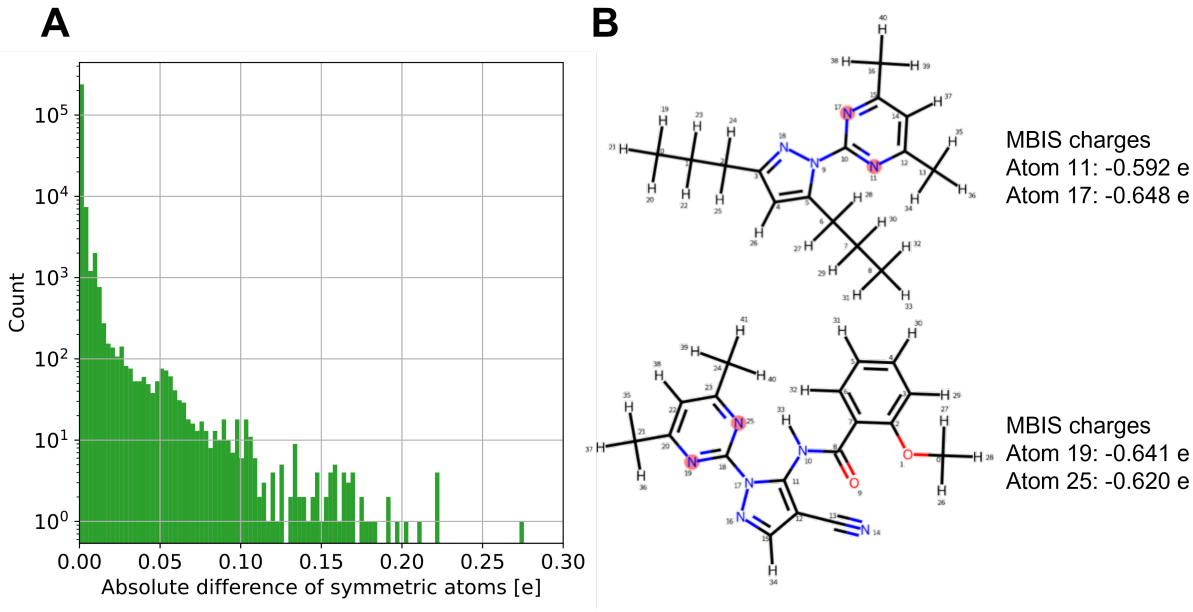

Figure S5: (**A**): Histogram of the difference between partial charges of topologically symmetric atoms in molecules of the validation set. (**B**): Example molecules with symmetric atoms (highlighted in red) that have asymmetric partial charges.

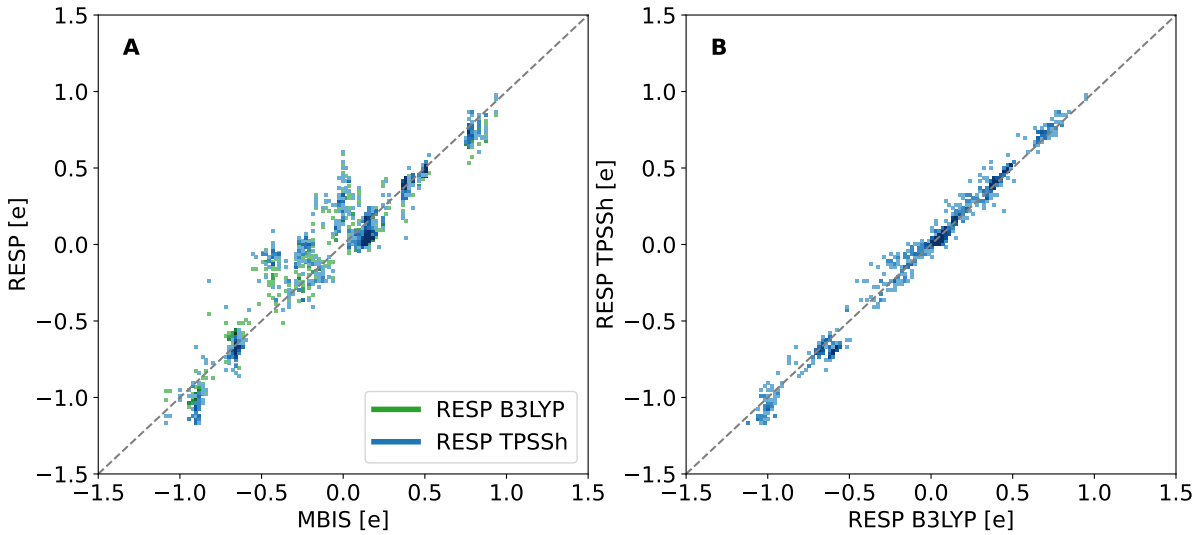

Figure S6: (**A**): Comparison of RESP2 charges for the amino acid test set calculated with two different levels of theory: B3LYP/sto-3g (green, commonly used default) and TPSSh/def2-TZVP (blue). (**B**): Comparison of the two different RESP charges to each other.

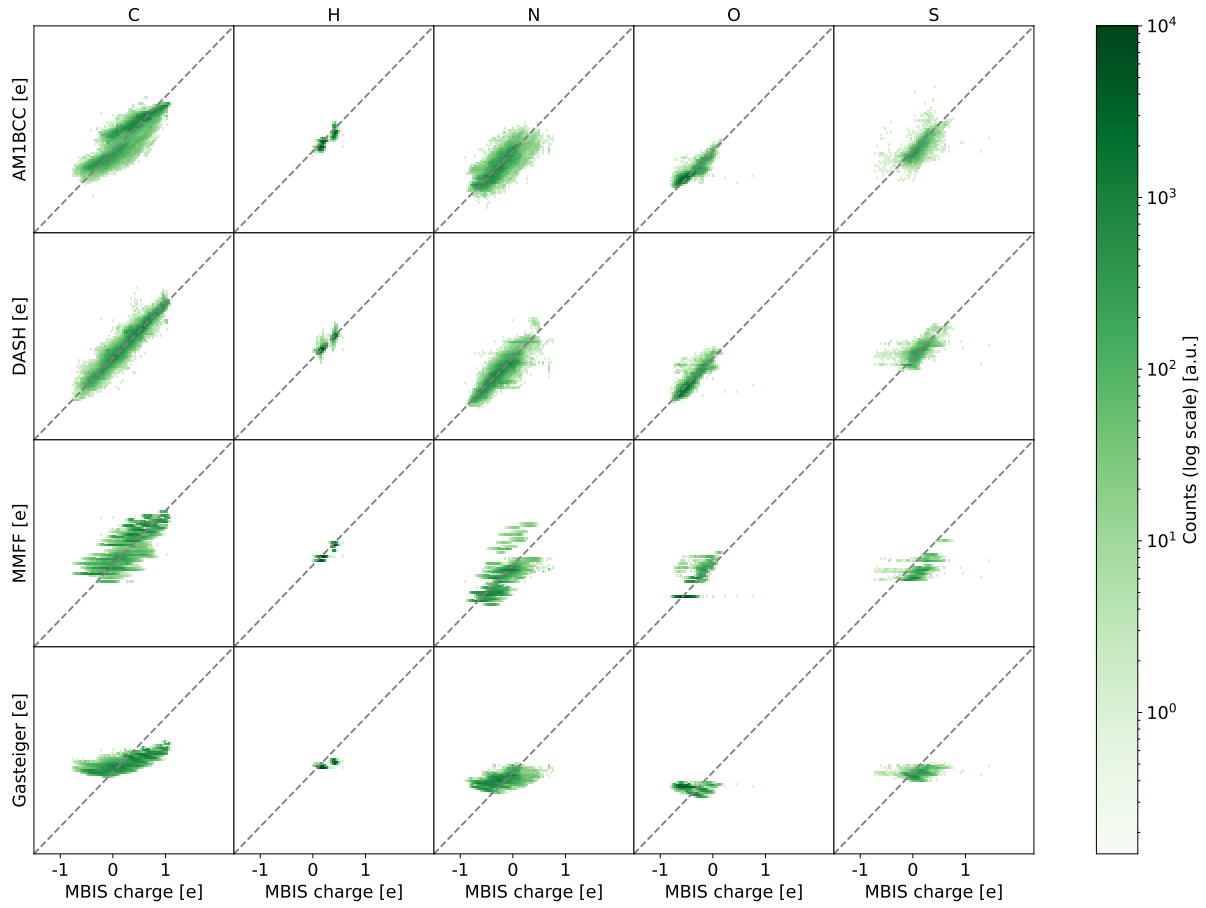

Figure S7: Four different charge models (AM1-BCC, DASH, MMFF94, Gasteiger) against the MBIS reference charges for the external VEHICLe[**?**] test set, shown per element (C, H, N, O, S).

Additional Tables

Table S1: Comparison of the training settings (learning rate and batch size) for the GNN in terms of RMSE and standard deviation (Std) on the validation set. The optimal values are highlighted in bold.

Learning rate	Batch size	RMSE [e]	Std [e]
0.000001	128	0.0198	0.00017
	256	0.0203	0.00010
	512	0.0213	0.00021
	64	0.0193	0.00015
0.00001	128	0.0175	0.00024
	256	0.0184	0.00046
	512	0.0188	0.00029
	64	0.0165	0.00021
0.0001	128	0.0153	0.00019
	256	0.0155	0.00014
	512	0.0159	0.00026
	64	**0.0153**	**0.00019**
0.001	128	0.0153	0.00016
	256	0.0153	0.00014
	512	0.0153	0.00022
	64	0.0153	0.00017
0.01	128	0.1814	0.02719
	256	0.1693	0.09596
	512	0.1523	0.05584
	64	0.2542	0.09108