A Related work

ML for chemical predictions. Computational methods for chemistry have a long history of applications. While the first methods for chemical reaction predictions relied mostly on hand-crafted symbolic rules [Salatin and Jorgensen, 1980], in recent years the computational the community interests shifted towards the use of deep learning techniques and derivatives. Nowadays, deep learning is used for a big variety of chemically related tasks, such as molecular graph generations [Sousa *et al.*, 2021; Goyal *et al.*, 2020; Bacciu and Podda, 2021; Simonovsky and Komodakis, 2018; Mercado *et al.*, 2021], molecule optimisation [Jin *et al.*, 2018; Griffiths and Hernández-Lobato, 2020; Korovina *et al.*, 2020], and chemical reaction predictions. Focusing on the latter, we can find two related approaches: forward prediction, where the goal is to infer the product molecules given a set of reactants, and retrosynthesis, a specular problem to forward synthesis consisting into predict the original reactants given a final product.

DL for chemical predictions. Initial attempts to use use deep learning models for reaction prediction used neural network to first identify the reaction centre in the reactants molecules [Coley *et al.*, 2019] or pairwise reactivity of atom bonds [Jin *et al.*, 2017; Fooshee *et al.*, 2018]. The proposals of the models were then used to algorithmically construct the final product of the reaction. A relevant collection of works aim at generating the whole reaction trees by iteratively selecting predefined reaction templates [Bradshaw *et al.*, 2018; Segler *et al.*, 2018; Bradshaw *et al.*, 2020; Gao *et al.*, 2021; Dai Nguyen and Tsuda, 2021; Nguyen and Tsuda, 2022]. Reaction templates are helpful for preventing the models to generate incorrect molecules that violates chemical rules, however they tend to be too restrictive in general, making it more difficult for the models to generalise to new types of reactions.

In the attempt to overcome the limitation of fixed reaction templates, it was proposed to combine powerful Transformer-based [Vaswani *et al.*, 2017] language models with the text SMILES representations of molecules in order to generate reaction outcomes in one step. These models leverage the flexibility of their architecture to directly learn a mapping between the reactants and the products. The first work of this kind is the Molecular Transformer [Schwaller *et al.*, 2019], able to beat the previous state of the art on both forward and backward prediction. Further improvements quickly followed with the Augmented Transformer model [Tetko *et al.*, 2020]. In this work, the authors focused on augmenting the original dataset by computing different variations of SMILES strings representing the same molecule.

The current top-1 accuracy state of the art for text-based, template-free models is represented by the Chemformer [Irwin *et al.*, 2022], where the authors use a self-supervised pre-training on a large variety of known molecules in order to increase performance on the reaction prediction downstream task. An interesting line of research tries to leverage the natural graph structure of molecules. [Tu and Coley, 2021] combine Transformers with Deep Graph Networks [Bacciu *et al.*, 2020] to get better initial molecular embeddings, while [Tavakoli *et al.*, 2022] build an hypergraph representation for the reaction by augmenting the disconnected molecular graph with "hypernodes" that represent molecules and reactions sides as a whole.

Finally, an orthogonal approach to chemical reaction prediction tries to leverage natural chemical constraints for ensuring the generation of chemically-plausible reactions. [Bradshaw *et al.*, 2018] model low-level reaction mechanisms for linear electron flow heterolythic reactions. This model generate the final products by predicting the linear electron flow on the initial set of reactants. [Qian *et al.*, 2020] combine neural networks with integer level programming constraints for expressing simple chemical constraints. Other works try to predict simplified version of the actual reactions mechanisms (often called "pseudo-mechanisms"), either via auto-regressively generating edits to the molecular graphs [Sacha *et al.*, 2021] or single-step predicting all possible bonds formation and deletion using a multi-pointer decoding network [Bi *et al.*, 2021]. The architectural constraints of these models generally ensure that the products resulting from a reaction are sound from a chemical perspective.

Benchmarking and studying logical and algebraic reasoning. After the recent impressive results of large scale pre-trained Transformers, several works started to investigate the reasoning abilities of these models [Geirhos *et al.*, 2020; Helwe *et al.*, 2021; Tran *et al.*, 2021]. [Wang *et al.*, 2021] show that large scale language models can only generalise well when the test distribution is the same of the training distribution, while they struggle in cross-distribution and out-of-distributions scenarios. [Liu *et al.*, 2020] introduce a dataset based on natural language multiple-choice questions. Each question requires a certain amount of logical reasoning in order to reach the correct answer. The baseline results show that the performance of current language models is still far behind human beings. The out-of-distribution problem is also highlighted in [Razeghi *et al.*, 2022], showing that the accuracy of these models on a certain training item is proportion to the number of times that that item as been seen in during training.

B Additional Results and Details from Section 3

The USPTO-T1 and USPTO-T2 variations are built in the following way:

- **USPTO-T1**: we duplicate all the molecules on their respective sides of the reaction (double reactants, and double products).
- **USPTO-T2**: for each reaction, we randomly select a molecule from either the reactants or the products. We then replicate the selected molecule on both sides of the reaction.

Please note that, with USPTO data, there is no a straightforward way to represent stoichiometry: in order to represent a double molecule in the reaction, we explicitly need to copy it twice in the data. Due to architectural limitations of the considered models, we are constrained to perform only a limited amount of augmentations (e.g. it would be impossible to replicate the entire reactants more than twice without exceeding the maximum sequence length). Table 5 report, as an

example, the Sabatier reaction re-adapted to the BAL, T1 and T2 variants.

For the experiments, we consider the following Transformer-based state-of-the-art models:

- Molecular Transformer (MOL.T) [Schwaller *et al.*, 2019]: the first transformer-based model that framed the problem of reaction predictions as sequence-to-sequence text translation.
- Augmented Transformer (AUG.T) [Tetko *et al.*, 2020]: extension of the Molecular Transformer, it was trained using different data augmentation schemes in order to boost its performance.
- Chemformer (CHEMF) [Irwin *et al.*, 2022]: it yields the current state of the art of top-1 accuracy on the reaction prediction tasks based on USPTO.
- Graph2SMILES (G2S) [Tu and Coley, 2021]: a Transformer-based model that uses a graph-based neural encoder for learning more expressive representations of the input molecules. The authors of G2S presents two versions (*dgcn* and *dgat*) of the model employing a different type of neural encoder.

Tables 6, 7, 8 ad 9 report the detailed results of these model over the USPTO, USPTO-BAL, USPTO-T1 and USPTO-T2 variants. We compute the following metrics:

- *Validity rate* (VAL): ratio of predictions that have a correct molecular structure.
- *Accuracy* (ACC): ratio of correct predictions.
- *At-least-one-accuracy* (ALO): we define a prediction to be "at-least-one accurate" if it contains all the ground truth molecules at least once. It can therefore be seens as a "molecule-only" accuracy, disregarding the multiplicity of the molecules.
- *Balanced predictions rate* (BAL): ratio of predictions that are balanced (i.e. that contains the same atoms as in the reactants).
- *Deficitary predictions rate* (DEF): ratio of predictions that do not contain atoms than are present in the reactants.
- *Exceeding predictions rate* (EXC): ratio of predictions that contain atoms that are not present in the reactants.
- *Deficitary and exceeding prediction rate* (D+E): ratio of prediction that are both exceeding and deficitary.

The table shows that, while the number of valid and accurate predictions is over 90% for all models, the balanced predictions are just a minority, less than 10%. Most of the unbalanced predictions are of the deficitary type, while a relative minority is of the exceeding type. One of the causes for the bad performance can be identified in the training data: as shown in Sec. 3.1, many of the reactions contained in USPTO are already unbalanced. This unbalance is reflected on the outputs of these models, another sign that these models are relying on memorising the data rather than trying to actually retrieve the correct underlying chemical mechanisms.

C Complete Algorithm for Building the Type-2 Stoichiometric Augmentations

Let $\mathcal{M}$ be the set of molecules in the reaction, we first sample as before a random coefficient $k_m \in [1, 5]$ for each molecule $m \in \mathcal{M}$. We then compute

$$\hat{k} = \min_{m \in \mathcal{M}} k_m. \tag{8}$$

The final coefficients are assigned as follows: on the reactants side of the reaction, we have

$$\forall m \in \mathcal{M}. \quad k_m = \begin{cases} k_m & \text{if } m \text{ is a reactant or a reagent,} \\ (k_m - \hat{k}) & \text{if m is a product, and } (k_m - \hat{k}) > 0\,, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Similarly, on the products side, we have

$$\forall m \in \mathcal{M}. \quad k_m = \begin{cases} k_m & \text{if } m \text{ is a product or a reagent,} \\ (k_m - \hat{k}) & \text{if m is a reactant, and } (k_m - \hat{k}) > 0\,, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

This assignment strategy takes into account the fact that specific quantities of reactants and reagents molecules are needed in order to get the respective products. The excess amount of molecules is just copied on the other side (a specular argument can be done for products). Please note that, following this algorithm, the new added molecules are chosen among the molecules that are already present in the original reaction.

D Additional details about the model and training procedure

We train a Transformer model [Vaswani *et al.*, 2017] with 6 encoder and 6 decoder layers. The number of attention heads is set to 8. Both the embeddings size, the internal model size and the size of the fully connected layers is set to 512. The loss function used is the cross-entropy between the predicted and the target strings.

We use a batch size of 64 for all experiments. the Adam optimizer is used for training, with an initial learning rate of 10^{-5}. The learning rate is halved when no loss improvement over the validation set is observed for 1000 steps, while the training is stopped after no loss improvement over the validation set is observed for 5000 steps. We use label smoothing with smoothing parameter 0.1.

For evaluation, we auto-regressively generate the model's prediction one character at a time, stopping the generation either when the model predicts an "end sequence" token or after 200 timesteps. The model is implemented and trained using the *pytorch_lightning* library.

E Multilabel Metrics for Evaluation

In ChemAlgebra, the model needs to predict a bag of product molecules, given an initial bag of reactants. This problem can be framed as a multilabel classification task, since multiple ground truth molecules (and their stoichiometric coefficients) have to be predicted at the same time. The computation of the usual classification metrics (such as accuracy, precision,

Table 5: Example of the the Sabatier reaction adapted to fit the BAL, T1 and T2 variations of the original USPTO. In T1, the reactants and products are doubled. In T2, one molecule (CO_2) is added on both side of the reaction.

	Reactants	Products
Original	$CO_2 + H_2$	$CH_4 + H_2O$
BAL	$CO_2 + H_2 + H_2 + H_2 + H_2 + Ni$	$CH_4 + H_2O + H_2O + Ni$
T1	$CO_2 + CO_2 + H_2 + H_2 + Ni$	$CH_4 + CH_4 + H_2O + H_2O + Ni$
T2	$CO_2 + CO_2 + H_2 + Ni$	$CO_2 + CH_4 + H_2O + Ni$

Table 6: Results of predictions of state-of-the-art models on the USPTO dataset.

	VAL	ACC	BAL	DEF	EXC	D+E
MOL.T	100.0	90.4	9.06	90.86	0.28	0.21
AUG.T	99.97	91.1	9.44	90.40	0.59	0.43
CHEMF	87.20	92.8	6.71	76.33	48.28	31.33
Graph2SMILES (dgcn)	99.98	90.3	8.83	90.62	1.60	1.06
Graph2SMILES (dgat)	99.98	90.3	8.88	90.52	1.67	1.08

Table 7: Results on predictions of state-of-the-art models on USPTO-BAL.

	VAL	ACC	BAL	DEF	EXC	D+E
MOL.T	99.85	1.39	1.50	98.48	0.30	0.29
CHEMF	78.20	0.0	4.51	92.39	22.52	19.43
G2S (dgat)	99.89	1.37	1.48	98.50	0.15	0.14
G2S (dgcn)	99.85	1.40	1.46	98.53	0.16	0.15

Table 8: Results on predictions of state-of-the-art models on USPTO-T1 dataset.

	VAL	ACC	ALO	BAL	DEF	EXC	D+E
MOL.T	99.03	0.04	56.66	0.18	99.76	1.41	1.37
CHEMF	91.88	0.23	34.41	0.29	99.22	6.31	5.83
G2S (dgat)	99.60	0.0	83.02	0.01	99.97	0.67	0.66
G2S (dgcn)	99.67	0.0	84.21	0.01	99.97	0.64	0.64

Table 9: Results of predictions of state-of-the-art models on USPTO-T2 dataset.

	VAL	ACC	ALO	BAL	DEF	EXC	D+E
MOL.T	99.03	0.03	28.32	0.38	99.08	5.76	5.23
CHEMF	67.42	0.0	2.61	2.48	90.00	25.06	17.55
G2S (dgat)	99.66	0.005	40.79	0.06	99.88	1.35	1.30
G2S (dgcn)	99.71	0.0025	41.43	0.04	99.90	1.19	1.14

recall, etc.) in the multilabel setting is not trivial, as we need to take into account different failure modes of the model at the same time. For example, the model could either predict the wrong molecule, or the right molecule but with a higher/lower coefficient than expected. For our tasks, we consider the Exact Match (EM), Jaccard (JAC) and F1 scores as multilabel metrics:

$$\mathrm{EM} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}(y_i = \hat{y}_i), \tag{11}$$

$$\mathrm{JAC} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|}, \tag{12}$$

$$\mathrm{F1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2|y_i \cap \hat{y}_i|}{|y_i| + |\hat{y}_i|}, \tag{13}$$

where y_i and $\hat{y}_i$ are the k-hot binarized vectors of the ground truth and the prediction, respectively. N is the number of test samples and $\mathbf{I}$ is the indicator variable checking for exact equality between y_i and $\hat{y}_i$.

Due to computational efficiency reasons, we do not explicitly binarize the vectors. Instead, we compute the number of true positives, false positives and false negatives by comparing the coefficients of corresponding molecules in both the prediction and the ground truth.

For example, if the model predicts $3\,H_2O + 2\,HCl + CO_2$ while the ground truth is $2\,H_2O + 2\,HCl + CH_4$, we count 4 true positives (two H_2O and two HCl), 2 false positives (one excess H_2O and one CO_2), and 1 false negative (the CH_4 in the ground truth). These counts are then combined to get the final multilabel metrics.

F Additional baseline results (molecule-only accuracy)

Table 10 contains the baseline results of the CHEMALGEBRA benchmarks, considering only the molecular-level accuracy (i.e. taking only into account the prediction of the correct molecules, disregarding stoichiometric coefficients). Using the same example of Appendix E, where the model predicts $3\,H_2O + 2\,HCl + CO_2$ while the ground truth is $2\,H_2O + 2\,HCl + CH_4$, we now count 2 true positives (the H_2O and HCl molecules), 1 false positives (CO_2), and 1 false negative (the CH_4 in the ground truth). In practice, here we are measuring only the predicted molecular structures, disregarding the stoichiometric coefficients.

We can observe that, for Type 1 tasks, the performance are very similar to Table 4. This is not surprising, as the model can easily learn to copy the same coefficient to the output, leaving only the task of molecular graph prediction to be learned. On the other hand, the performance on Type 2 variants are higher than Table 4. This can be explained as the main challenge of Type 2 tasks is the prediction of the correct coefficients, which is disregarded by these "molecule-only" metrics. The consequence of the different choice of metrics can be also observed in the out-of-distribution performance, that in Table 10 is consistently higher than Table 4: not considering the stoichiometric coefficients makes the out-of-distribution task very similar to a standard reaction prediction tasks (with the out-of-distribution coefficients acting as confounding tokens).

G Overview of the CHEMALGEBRA variants

Table 11 contains an overview of the different CHEMALGEBRA variants that are described in Section 4. Each variant is contained in a separated folder. The filenames are given according to the following conventions:

- Files with prefix "`src`" contain the input molecules, while files with prefix "`tgt`" contain the target molecules.
- Substrings "`train`", "`valid`" or "`test`" refer to the respective subsets of data to be used during cross-validation.
- Test files have the additional suffixes "`_in`" or "`_out`", referring respectively to the in-distribution test set and out-of-distribution test set, as described in Section 4.
- Files with the "`_cross`" suffix refer to the cross-distribution setting described in Section 4. Note that the "`_cross`" tasks have their own dedicated training and validation sets.

In order to provide a visual intuition of the actual data contained in CHEMALGEBRA, we show in Tables 12, 13, 14, 15, 16, 17, 18 and 19 the first five training reactions for each CHEMALGEBRA variant.

Table 10: Jaccard (JAC) and F1-score (F1) on the CHEMALGEBRA datasets.

	test-in		test-cross		test-out	
	JAC	F1	JAC	F1	JAC	F1
FORMULA_T1x1	63.03	72.43	67.39	75.71	23.96	35.79
FORMULA_T1x5	62.09	71.69	60.64	70.59	26.03	38.11
FORMULA_T2x1	48.89	61.48	35.10	48.44	26.35	39.27
FORMULA_T2x5	50.33	62.80	36.29	49.54	27.25	40.26
SMILES_T1x1	44.43	57.49	45.74	58.54	24.44	36.52
SMILES_T1x5	41.85	54.99	42.00	55.17	24.50	36.28
SMILES_T2x1	27.63	40.57	24.64	36.97	18.03	28.45
SMILES_T2x5	28.47	41.54	24.56	36.98	14.59	23.50

Table 11: Summary of all eight variants of CHEMALGEBRA. "Name" is the name of the variant in the dataset's files. The "Representation" column specifies which representation is used for the reactions of the dataset. The "Task Type" column determines the choice of coefficients (according to the reasoning tasks defined in Section 2.1). The "Augmentation" column specifies how many stoichiometric assignments are sampled for each original USPTO reaction in the training set.

Name	Representation	Task Type	Augmentations
`ChemAlgebra_F_T1_x1`	Chemical formula	Type 1	x1
`ChemAlgebra_F_T1_x5`	Chemical formula	Type 1	x5
`ChemAlgebra_F_T2_x1`	Chemical formula	Type 2	x1
`ChemAlgebra_F_T2_x5`	Chemical formula	Type 2	x5
`ChemAlgebra_S_T1_x1`	SMILES string	Type 1	x1
`ChemAlgebra_S_T1_x5`	SMILES string	Type 1	x5
`ChemAlgebra_S_T2_x1`	SMILES string	Type 2	x1
`ChemAlgebra_S_T2_x5`	SMILES string	Type 2	x5

Table 12: Sample of training chemical reactions of `ChemAlgebra_F_T1_x1`.

INPUT	TARGET
{5}C7H4ClNO4.{5}H2O.{5}CH5N	{5}HCl.{5}H2O.{5}C8H8N2O4
{5}C7H7Cl2N.{5}C11H5Cl2N3S	{5}HCl.{5}C18H11Cl3N4S
{1}H.{1}HCl.{1}H2O.{1}C15H10ClFN4.{1}C2H4O2	{1}C2H4O2.{1}HCl.{1}CN.{1}H2O.{1}C14H11ClFN3
{4}C4H8O.{4}H.{4}C26H33N3O3.{4}Na+.{4}Na+. {4}H2O4S.{4}CH2O.{4}CHO3-.{4}H4B-	{4}C27H35N3O3.{4}O3-.{4}C4H8O.{4}Na+. {4}H4B-.{4}H2O4S.{4}Na+.{4}CH2O
{4}C5H5N.{4}C5H4ClNO2.{4}HCl.{4}C6H4ClNO	{4}C5H5N.{4}HCl.{4}C11H7ClN2O3.{4}HCl

Table 13: Sample of training chemical reactions of `ChemAlgebra_F_T1_x5`.

INPUT	TARGET
{3}C7H4ClNO4.{3}CH5N.{3}H2O	{3}HCl.{3}C8H8N2O4.{3}H2O
{3}CH5N.{3}C7H4ClNO4.{3}H2O	{3}H2O.{3}HCl.{3}C8H8N2O4
{3}H2O.{3}C7H4ClNO4.{3}CH5N	{3}C8H8N2O4.{3}H2O.{3}HCl
{2}H2O.{2}C7H4ClNO4.{2}CH5N	{2}C8H8N2O4.{2}HCl.{2}H2O
{3}C7H4ClNO4.{3}CH5N.{3}H2O	{3}HCl.{3}C8H8N2O4.{3}H2O

Table 14: Sample of training chemical reactions of `ChemAlgebra_F_T2_x1`.

INPUT	TARGET
{4}C7H4ClNO4.{2}C8H8N2O4.{2}H2O.{3}CH5N	{2}H2O.{3}C8H8N2O4.{3}C7H4ClNO4.{2}CH5N.{1}HCl
{5}C11H5Cl2N3S.{1}HCl.{4}C7H7Cl2N.{1}C18H11Cl3N4S	{2}HCl.{4}C11H5Cl2N3S.{2}C18H11Cl3N4S.{3}C7H7Cl2N
{2}H2O.{3}H.{2}C15H10ClFN4.{3}CN. {2}C2H4O2.{3}C14H11ClFN3.{5}HCl	{2}H.{4}CN.{2}C2H4O2.{2}H2O. {1}C15H10ClFN4.{5}HCl.{4}C14H11ClFN3
{3}C26H33N3O3.{3}CHO3-.{3}H2O4S.{1}C4H8O.{5}H4B-. {5}Na+.{5}Na+.{4}H.{4}C27H35N3O3.{1}CH2O	{1}C4H8O.{1}O3-.{5}Na+.{1}CH2O.{3}H2O4S.{2}CHO3-. {5}Na+.{5}H4B-.{2}C26H33N3O3.{3}H.{5}C27H35N3O3
{2}C5H4ClNO2.{3}C5H5N.{3}HCl.{1}C6H4ClNO.{3}C11H7ClN2O3.{4}HCl	{4}C11H7ClN2O3.{3}C5H5N.{1}C5H4ClNO2.{4}HCl.{4}HCl

Table 15: Sample of training chemical reactions of ChemAlgebra_F_T2_x5.

INPUT	TARGET
{1}C8H8N2O4.{4}H2O.{2}C7H4ClNO4.{2}CH5N.{2}HCl	{4}H2O.{4}HCl.{3}C8H8N2O4
{3}C8H8N2O4.{2}C7H4ClNO4.{3}H2O.{2}HCl.{5}CH5N	{3}CH5N.{5}C8H8N2O4.{4}HCl.{3}H2O
{3}C7H4ClNO4.{5}CH5N.{2}H2O.{4}C8H8N2O4	{2}C7H4ClNO4.{1}HCl.{4}CH5N.{2}H2O.{5}C8H8N2O4
{2}CH5N.{1}C7H4ClNO4.{3}HCl.{2}C8H8N2O4.{5}H2O	{1}CH5N.{3}C8H8N2O4.{5}H2O.{4}HCl
{1}H2O.{2}CH5N.{1}HCl.{1}C7H4ClNO4	{1}H2O.{1}C8H8N2O4.{1}CH5N.{2}HCl

Table 16: Sample of training chemical reactions of ChemAlgebra_S_T1_x1.

INPUT	TARGET
{4}CN.4O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{4}O	{4}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{4}Cl.{4}O
{1}NCc1ccc(Cl)c(Cl)c1.{1}Clc1cc2c(Cl)nc(-c3ccncc3)nc2s1	{1}Cl.{1}Clc1cc2c(NCc3ccc(Cl)c(Cl)c3)nc(-c3ccncc3)nc2s1
{4}Cc1c(Cl)nnc(C(C#N)c2ccc(F)c(C#N)c2)c1C.{4}Cl.{4}O. {4}CC(=O)O.{4}[H]	{4}O.{4}Cl.{4}Cc1c(Cl)nnc(Cc2ccc(F)c(C#N)c2)c1C. {4}[C][N].{4}CC(=O)O
{5}[BH4-].{5}O=S(=O)(O)O.{5}C=O.{5}C1CCOC1. {5}[Na+].{5}O=C([O-])O.{5}[Na+]. {5}CC(C)(C)OC(=O)c1ccc(NCC2(O)CCN(CCc3ccc(C#N)cc3)CC2)cc1. {5}[H]	{5}O=S(=O)(O)O.{5}[Na+].{5}C=O.{5}[BH4-]. {5}CN(CC1(O)CCN(CCc2ccc(C#N)cc2)CC1)c1ccc(C(=O)OC(C)(C)C)cc1. {5}[Na+].{5}[O]O[O-].{5}C1CCOC1
{3}O=c1cc(O)c(Cl)c[nH]1.{3}Cl.{3}O=C(Cl)c1cccnc1. {3}c1ccncc1	{3}O=C(Oc1cc(=O)[nH]cc1Cl)c1cccnc1.{3}Cl. {3}c1ccncc1.{3}Cl

Table 17: Sample of training chemical reactions of ChemAlgebra_S_T1_x5.

INPUT	TARGET
{4}O.{4}CN.{4}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1	{4}O.{4}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{4}Cl
{2}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{2}CN.{2}O	{2}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{2}Cl.{2}O
{4}CN.{4}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{4}O	{4}O.{4}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{4}Cl
{5}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{5}O.{5}CN	{5}Cl.{5}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{5}O
{3}O.{3}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{3}CN	{3}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{3}O.{3}Cl

Table 18: Sample of training chemical reactions of ChemAlgebra_S_T2_x1.

INPUT	TARGET
{3}CNc1ccc(C(=O)O)cc1[N+](=O)[O-]. {5}CN.{4}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{1}Cl.{3}O	{4}CNc1ccc(C(=O)O)cc1[N+](=O)[O-]. {3}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{4}CN.{3}O.{2}Cl
{4}NCc1ccc(Cl)c(Cl)c1.1Cl. {3}Clc1cc2c(Cl)nc(-c3ccncc3)nc2s1	{2}Cl.1Clc1cc2c(NCc3ccc(Cl)c(Cl)c3)nc(-c3ccncc3)nc2s1. {3}NCc1ccc(Cl)c(Cl)c1.2Clc1cc2c(Cl)nc(-c3ccncc3)nc2s1
{1}[H].1Cc1c(Cl)nnc(Cc2ccc(F)c(C#N)c2)c1C. {3}Cc1c(Cl)nnc(C(C#N)c2ccc(F)c(C#N)c2)c1C. {1}O.{5}CC(=O)O.{1}Cl	{1}Cl.{5}CC(=O)O.1O.1[C][N]. {2}Cc1c(Cl)nnc(Cc2ccc(F)c(C#N)c2)c1C. {2}Cc1c(Cl)nnc(C(C#N)c2ccc(F)c(C#N)c2)c1C
{4}[H].{3}[Na+].{2}O=S(=O)(O)O.{5}C1CCOC1.{1}C=O.{3}[Na+]. {4}CC(C)(C)OC(=O)c1ccc(NCC2(O)CCN(CCc3ccc(C#N)cc3)CC2)cc1. {2}O=C([O-])O.{2}[BH4-]	{2}[BH4-].{3}[Na+].{1}C=O.{2}O=S(=O)(O)O.1[O]O[O-]. {3}[H].1O=C([O-])O.{5}C1CCOC1. {1}CN(CC1(O)CCN(CCc2ccc(C#N)cc2)CC1)c1ccc(C(=O)OC(C)(C)C)cc1. {3}CC(C)(C)OC(=O)c1ccc(NCC2(O)CCN(CCc3ccc(C#N)cc3)CC2)cc1. {3}[Na+]
{2}c1ccncc1.{3}O=C(Cl)c1cccnc1.{3}Cl.{4}Cl. {4}O=C(Oc1cc(=O)[nH]cc1Cl)c1cccnc1.{1}O=c1cc(O)c(Cl)c[nH]1	{2}c1ccncc1.{2}O=C(Cl)c1cccnc1.{4}Cl.{4}Cl. {5}O=C(Oc1cc(=O)[nH]cc1Cl)c1cccnc1

Table 19: Sample of training chemical reactions of ChemAlgebra_S_T2_x5.

INPUT	TARGET
{3}CN.{5}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1. {3}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{4}O	{2}CN.{4}CNc1ccc(C(=O)O)cc1[N+](=O)[O-]. {4}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{1}Cl.{4}O
{1}CNc1ccc(C(=O)O)cc1[N+](=O)[O-]. {5}CN.{5}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{5}O	{2}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{4}CN.{1}Cl. {5}O.{4}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1
{5}O.{2}Cl.{2}CN.{4}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1. {1}CNc1ccc(C(=O)O)cc1[N+](=O)[O-]	{3}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{1}CN. {3}Cl.{2}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{5}O
{1}CN.{4}CNc1ccc(C(=O)O)cc1[N+](=O)[O-]. {3}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{3}O	{1}Cl.{3}O.{5}CNc1ccc(C(=O)O)cc1[N+](=O)[O-]. {2}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1
{2}O.{5}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1.{5}CN.{2}Cl	{1}CNc1ccc(C(=O)O)cc1[N+](=O)[O-].{2}O.{3}Cl. {4}CN.{4}O=C(O)c1ccc(Cl)c([N+](=O)[O-])c1