Table 2: Hyperparameters for SFT model training. These are fixed across all dataset splits and model sizes and types for summarisation.

Hyperparameter	Value
batch size	128
epochs	1
adam beta1	0.9
adam beta2	0.999
adam epsilon	1e-8
frozen layers	80%

Table 3: Hyperparameters for RM training. These are fixed across all dataset splits and model sizes and types for summarisation.

Hyperparameter	Value
batch size	64
epochs	1
adam beta1	0.9
adam beta2	0.999
adam epsilon	1e-8
frozen layers	80%

E.3 Model selection

For the results reported in the paper, we sweep over 3-5 learning rates for each model type, and select the best model on the validation set using an appropriate metric (accuracy for RMs, loss for SFT, reward for RL) - see Appendix E.4. For both the in-distribution and out-of-distribution results, we always use a validation set drawn from the same distribution as training. This more closely matches a real-world setting where we would not have access to OOD data to do model selection, and would have to draw model selection data from the same distribution used in training.

E.4 Hyperparameters

Summarisation. For each model type (SFT, RM, RLHF) we do a sweep over learning rate, choosing ranges of values informed by choices in previous work (Stiennon et al., 2022) and early experimentation. The results in the paper are the best model with the learning rate chosen on an in-distribution validation set using loss, accuracy and reward respectively for SFT, RM and RLHF training. The learning rates for SFT are 3e-4, 1e-4, 3e-5, with 3e-5 selected; for RMs are 3e-4, 1e-4, 3e-5, 1e-5, 3e-6, with 3e-5 selected; for RLHF are: 1.5e-6, 3e-6, 6e-6, 1.5e-5, 3e-5, with 1.5e-5 selected.

We list the other hyperparameters (which are unchanged between all runs) for SFT, RM and RLHF training in Table 2, Table 3 and Table 4 respectively. We chose these following prior work (Stiennon et al., 2022).

Instruction Following. For the instruction following results, we use the models released by (Dubois et al., 2023), and so the hyperparameters can be found in that work.

F Differences from Stiennon et al. (2022)

As we mostly follow (Stiennon et al., 2022) in the training of our summarisation models, we here describe the main differences between our work and theirs in terms of training. For RLHF, we train a single model with policy and value head, rather than separate policy and value functions. This is much more computationally efficient, and follows other recent work that still achieves impressive results (Glaese et al., 2022). This means that we randomly initialise the value function head, rather than intialising the value function from the reward model as is done in (Stiennon et al., 2022). We use LLaMa (Touvron et al., 2023a) (and OPT (Zhang et al., 2022) in Appendix J) as our pretrained

Table 4: Hyperparameters for RLHF model training. These are fixed across all dataset splits and model sizes and types for summarisation. One PPO step consists of generating *batch size* samples, and then performing *ppo epochs* of optimization on them, split into *ppo minibatch size* minibatches.

Hyperparameter	Value
batch size	256
ppo epochs	4
ppo steps	750
ppo minibatch size	256
KL penalty coefficient	0.05
normalise advantages	True
adam beta1	0.9
adam beta2	0.999
adam epsilon	1e-8
frozen layers	80%

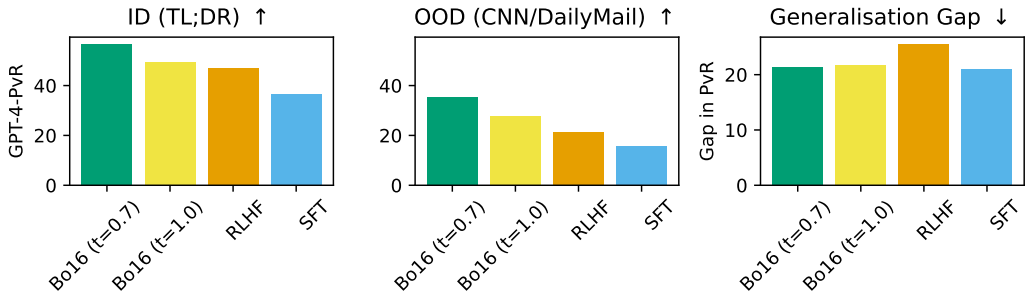

Figure 9: GPT-4 API evaluation win rate vs reference summaries for SFT, Bo16 with two different temperatures, and RL models, trained on the summarisation task. In-distribution is performance on TL;DR, and out-of-distribution is on CNN/DailyMail. Effectively a version of Fig. 2 with addition Bo16 results with a worse-performing temperature

models, while they use unreleased models which were trained in a similar way to GPT-3 (Brown et al., 2020).

We freeze the first 80% of the layers and the embedding and unembedding layers, as more recent work (Glaese et al., 2022; Menick et al., 2022) has shown that good results can still be achieved, and training is much more computationally efficient. In Table 14 we show the drop in performance for smaller OPT models between freezing 80% of the layers and not freezing. There is a drop of performance, but it is not catastrophic (equivalent to about a drop in model size among the three model sizes used), which justifies the tradeoff of training models with partially frozen weights.

G Best of N Temperature Experiment

In the main paper we report results for BoN using temperature 0.7. Here we show results for BoN with temperature 1, and show that temperature 0.7 performs better, hence our choice of it as the hyperparamter we use. Fig. 9 shows the results of BoN with two temperatures, as well as RLHF and SFT for comparison.

H Sequential Instructions Dataset Details

For the sequential instructions dataset, we build on the AlpacaFarm variant of the Self-Instruct protocol (Dubois et al., 2023; Wang et al., 2023), but adjust the seed instructions and prompt to gather more sequential instructions. Fig. 10 shows the prompt used to generate these instructions,

```
You are asked to come up with a set of 20 diverse task instructions.
These task instructions will be given to a GPT model and we will
evaluate the GPT model for completing the instructions.

Here are the requirements:
1. Try not to repeat the verbs for each instruction to maximize diversity.
2. The language used for the instruction also should be diverse. For
example, you should combine questions with imperative instrucitons.
3. The type of instructions should be diverse. The list should include
diverse types of tasks like open-ended generation, classification, editing,
etc.
4. A GPT language model should be able to complete the instruction. For
example, do not ask the assistant to create any visual or audio output.
For another example, do not ask the assistant to wake you up at 5pm or
set a reminder because it cannot perform any action.
5. The instructions should be in English.
6. The instructions should be a sequential or compositional instruction
containing multiple steps, where each step is related to the previous
steps. Either an imperative sentence or a question is permitted.
7. Try not to repeat the verbs used for each part of the instruction
across instructions to maximize diversity.
8. The output should be an appropriate response to the instruction and
the input. Make sure the output is less than 100 words.

List of 20 tasks:
```

Figure 10: The prompt for text-davinci-003 to produce instructions for the sequential instructions dataset using the Self-Instruct protocol (Wang et al., 2023).

Table 5: Example inputs from the sequential instructions dataset

Generate a list of items that may be found in a first aid kit, along with description on why each item is important.
Sort a list of emotions (sadness, joy, anger, fear, disgust) into two categories, and explain why each emotion fits into the categories created.
Explain the concept of Renormalization Group in simple words, describe three uses for Renormalization Group in theoretical physics, and discuss its relationship with scaling laws.
Summarize the history of the Cold War, explain the outcome of the war, and discuss its significance to the world today.

and Table 5 shows examples of generated from the dataset. This dataset is accessible here: `https://huggingface.co/datasets/UCL-DARK/sequential-instructions`.

I SUMMARISATION KL SWEEP RESULTS

Here we present results for generalisation and diversity for LLaMa models trained with RLHF on the summarisation task, sweeping over the KL penalty coefficient. This hyperparameter determines the weight of the KL penalty in the RLHF reward (the β_{KL} in Eq. (1)). It could be the case that this KL penalty can control the tradeoff between diversity and generalisation, and so we try multiple different values of the penalty and include the results here.

We found that higher KL penalties actually resulted in less output diversity (Figs. 12 and 13), and also generally worse performance (Fig. 11), showing that the choice of KL penalty did not seem to provide a way to tradeoff between diversity and generalisation.