A Synthetic Pre-training Details

We do synthetic pre-training with the same hyperparameters that the off-the-shelf pre-trained T5-small was trained with: AdaFactor optimizer, batch size 128, sequence length 512, and inverse square root learning rate $1/\sqrt{max(n, 10000)}$ where n is the current step. We evaluate token validation accuracy and save a checkpoint every 5000 steps. For synthetic tasks besides the artificial language, we fine-tuned with the first checkpoint that the model reaches above 99% token validation accuracy. For LIME, Nonsense Summary, Set, and Identity this was 30K steps, 5K steps, 10K steps, and 5K steps respectively. For the artificial language pre-training, after training for 300K steps the model's token validation accuracy had plateaued without converging to above 99% token validation accuracy, so we chose to fine-tune with the max accuracy checkpoint, which was at step 165K with accuracy of 77.7%. For reference, off-the-shelf pre-trained T5-small was trained for 524,288 steps (Raffel et al., 2020).

B Fine-tuning Details

We fine-tuned with the same hyperparameters (Raffel et al., 2020) fine-tuned with, with the exception of learning rate, for which we do a sweep over 1e-2, 3e-3, 1e-3. We use the AdaFactor optimizer and batch size 128 for all except WebQSP and Code Translation, for which we use batch size 32 due to their long sequence length. We list the learning rate and sequence length for each task in Appendix D.3. We use the default T5 tokenization (Raffel et al., 2020). We fine-tuned for a max of 150K steps for code translation, 400K steps for retrosynthesis, 250K steps for SQuAD, 50K steps for WebQSP, 100K steps for MTOP, and 100K steps for CNNDM-10K. We evaluate and save checkpoints every 10K steps for all tasks besides WebQSP, for which we evaluate and save checkpoints every 5K steps. For all tasks besides CNNDM-10K and SQuAD, we report the test set metric achieved with the checkpoint corresponding to the max token validation accuracy checkpoint. For CNNDM-10K, we reported the test set metric achieved at step 50K, because we observed the test set metric achieved with the max token validation accuracy checkpoint was often significantly worse than later step checkpoints, and the test set metric always plateaued before and up to about 50K steps. For SQuAD, we report the best validation set metric, as was done in (Raffel et al., 2020), because evaluating on the test set requires running inference on a benchmark server.

C Off-the-shelf Pre-trained T5v1.1-Small for CNNDM-10K

The differences between Pre-trained T5v1.1 and Pre-trained T5 are:

1. T5v1.1 Uses the GEGLU activation in feed-forward hidden layers rather than ReLU.
2. Dropout was turned off in pre-training for T5v1.1.
3. No parameter sharing between the embedding and classifier layer for T5v1.1.
4. **T5v1.1 was pre-trained on C4 only, without mixing in the downstream tasks.**

The last difference is why we fine-tune CNNDM-10K with Pre-trained T5v1.1 instead of Pre-trained T5. Using Pre-trained T5 would be an unfair comparison because it has already trained been on all 290K CNNDM task training data.

D Downstream Tasks

D.1 Task Descriptions

CNNDM-10K is 10K training examples from the CNNDM benchmark, which consists of news articles from CNN and Daily Mail and summaries of each article (See et al., 2017). We evaluated on CNNDM-10K instead of the full CNNDM because this was the dataset that (Krishna et al., 2021) evaluated their nonsense summarization synthetic task on, which was one of the synthetic tasks we evaluate in Section 3.

MTOP (Li et al., 2021) and **WebQSP** (Yih et al., 2016) are two benchmarks for semantic parsing, the task of converting a natural language query to a logical form. For these tasks, we use the data from (Xie et al., 2022) that is already processed to be in a Seq2Seq format compatible with T5. MTOP has 17K training examples and WebQSP has 2.7K training examples.

SQuAD 1.1 is a reading comprehension dataset consisting of "questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text or span from the corresponding reading passage" (Rajpurkar et al., 2016). There are 87K training examples.

Java to C# code translation is one task within Microsoft's CodeXGLUE: General Language Understanding Evaluation benchmark for Code (Lu et al., 2021). The task consists of 10K training examples that were collected from the code of several open-source projects that were originally developed in Java and then ported to C#.

USPTO-50K is a dataset of 50K chemical reactions that is a commonly used to benchmark deep learning methods for the chemistry task of single-step retrosynthesis, which is predicting possible reactants when given a product as input. (Liu et al., 2017). For this task, we use the data from (Lu & Zhang, 2022) that is already processed to be in a Seq2Seq format compatible with T5. There are 40K training examples.

D.2 Task Examples

```
Source:
summarize: marouane fellaini and adnan januzaj continue to show the world they are not just teammates but
      also best mates . the manchester united and belgium duo both posted pictures of themselves out at a
      restaurant on monday night ahead of their game against newcastle on wednesday . januzaj poses in the
      middle of fellaini and a friend looking like somebody who failed to receive the memo about it being a
       jackson 5 themed night . premier league duo adnan januzaj and marouane fellaini pose with a friend
      on the dance floor . manchester united and belgium duo fellaini and januzaj are good friends both on
      and off the pitch . manchester united ace fellaini runs over to the bench to celebrate his goal
      against qpr with friend januzaj . the disco effect in the background adds to the theory, but januzaj
      doesn't seem to mind as they later pose on the dance floor with other friends. united haven't had too
       many reasons to have a song and dance this season so it seems they may be hitting the discotheques
      as another form of release . however, victory against newcastle on wednesday would leave manager
      louis van gaal at least tapping his toes as they continue to fight for a champions league spot this
      season. januzaj and robin van persie join fellaini in celebrating in front of the manchester united
      fans at west brom . januzaj receives some words of wisdom from manchester united's dutch manager
      louis van gaal . januzaj and fellaini are joined by some friends as they take to the dance floor
      ahead of the newcastle game .

Target:
the belgian duo took to the dance floor on monday night with some friends . manchester united face
      newcastle in the premier league on wednesday . red devils will be looking for just their second
      league away win in seven . louis van gaal's side currently sit two points clear of liverpool in
      fourth .
```

```
Source:
Has Angelika Kratzer video messaged me ? ; structured knowledge: IN:GET: MESSAGE, WEATHER, ALARM, INFO_
      RECIPES, STORIES_NEWS, REMINDER, RECIPES, EVENT, CALL_TIME, LIFE_EVENT, INFO_CONTACT, CONTACT, TIMER,
       REMINDER_DATE_TIME, AGE, SUNRISE, EMPLOYER, EDUCATION_TIME, JOB, AVAILABILITY, CATEGORY_EVENT, CALL,
       EMPLOYMENT_TIME, CALL_CONTACT, LOCATION, TRACK_INFO_MUSIC, SUNSET, MUTUAL_FRIENDS, UNDERGRAD,
      REMINDER_LOCATION, ATTENDEE_EVENT, MESSAGE_CONTACT, REMINDER_AMOUNT, DATE_TIME_EVENT, DETAILS_NEWS,
      EDUCATION_DEGREE, MAJOR, CONTACT_METHOD, LIFE_EVENT_TIME, LYRICS_MUSIC, AIRQUALITY, LANGUAGE, GENDER,
       GROUP | IN:SEND: MESSAGE | IN:SET: UNAVAILABLE, RSVP_YES, AVAILABLE, DEFAULT_PROVIDER_MUSIC, RSVP_
      INTERESTED, DEFAULT_PROVIDER_CALLING, RSVP_NO | IN:DELETE: REMINDER, ALARM, TIMER, PLAYLIST_MUSIC |
      IN:CREATE: ALARM, REMINDER, CALL, PLAYLIST_MUSIC, TIMER | IN:QUESTION: NEWS, MUSIC | IN:PLAY: MUSIC,
      MEDIA | IN:END: CALL | IN:IGNORE: CALL | IN:UPDATE: CALL, REMINDER_DATE_TIME, REMINDER_TODO, TIMER,
      METHOD_CALL, ALARM, REMINDER_LOCATION, REMINDER | IN:PAUSE: MUSIC, TIMER | IN:ANSWER: CALL | IN:
      SNOOZE: ALARM | IN:IS: TRUE_RECIPES | IN:REMOVE: FROM_PLAYLIST_MUSIC | IN:ADD: TIME_TIMER, TO_
      PLAYLIST_MUSIC | IN:SHARE: EVENT | IN:PREFER:  | IN:START: SHUFFLE_MUSIC | IN:SILENCE: ALARM | IN:
      SWITCH: CALL | IN:SUBTRACT: TIME_TIMER | IN:PREVIOUS: TRACK_MUSIC | IN:HOLD: CALL | IN:SKIP: TRACK_
      MUSIC | IN:LIKE: MUSIC | IN:RESTART: TIMER | IN:RESUME: TIMER, CALL, MUSIC | IN:MERGE: CALL | IN:
      REPLAY: MUSIC | IN:LOOP: MUSIC | IN:STOP: MUSIC, SHUFFLE_MUSIC | IN:UNLOOP: MUSIC | IN:CANCEL:
      MESSAGE, CALL | IN:REWIND: MUSIC | IN:REPEAT: ALL_MUSIC, ALL_OFF_MUSIC | IN:FAST: FORWARD_MUSIC | IN:
      DISLIKE: MUSIC | IN:DISPREFER:  | IN:HELP: REMINDER | IN:FOLLOW: MUSIC

Target:
[IN:GET_MESSAGE [SL:CONTACT Angelika Kratzer ] [SL:TYPE_CONTENT video ] [SL:RECIPIENT me ] ]
```

```
Source:
where is isthmus of panama located? ; structured knowledge: Isthmus of Panama: m.04zwft | m.06n3y location.
     location.contains m.06w99sr | m.06n3y location.location.contains m.0cn1d6 | m.06n3y location.location.
     contains m.0c_ys | m.06n3y location.location.contains m.03718t | m.06n3y location.location.contains m
     .0w_hmnw | m.06n3y location.location.contains m.0c4rq4 | m.06n3y location.location.contains m.02t1x0
     | m.06n3y location.location.contains m.05w5vr | m.04zwft common.topic.notable_types m.01nt | m.06n3y
     location.location.contains m.09vgjp | m.06n3y location.location.contains m.0b2ft5 | m.06n3y location.
     location.contains m.07twz | m.06n3y location.location.contains m.0cqy3y | m.04zwft location.location.
     containedby m.06n3y | m.06n3y location.location.contains m.02x1y_b | m.06n3y location.location.
     contains m.03t3qb | m.06n3y location.location.contains m.06w57th | m.06n3y location.location.contains
      m.06w9q8y | m.06n3y location.location.contains m.0cpbxws | m.06n3y location.location.contains m.07t1
     sw | m.06n3y location.location.contains m.01nr2h | m.06n3y location.location.contains m.06gldq | m.06
     n3y location.location.contains m.034m8 | m.06n3y location.location.contains m.0dkz7x | m.06n3y
     location.location.contains m.0c7r4h | m.06n3y location.location.contains m.025vjdw | m.06n3y location.
     location.contains m.02qbjjz | m.06n3y location.location.contains m.0wq95z_ | m.06n3y location.
     location.contains m.0sd7 | m.06n3y location.location.contains m.06bf8s | m.06n3y location.location.
     contains m.027d7t4 | m.06n3y location.location.contains m.0cn1vs | m.06n3y location.location.contains
      m.076ycbb | m.06n3y location.location.contains m.061b7x | m.06n3y location.location.contains m.0p2n
     | m.06n3y location.location.contains m.0w_j2g5 | m.06n3y location.location.contains m.015fr | m.06n3y
      location.location.contains m.02xsr6 | m.06n3y location.location.contains m.02z722h | m.06n3y
     location.location.contains m.06w3qq8 | m.06n3y location.location.contains m.0cn1k1 | m.06n3y location.
     location.contains m.02vrgz0 | m.06n3y location.location.contains m.0g61h6 | m.06n3y location.location.
     contains m.0w_j7px | m.06n3y location.location.contains m.05fzj6 | m.06n3y location.location.contains
      m.026h5zg | m.06n3y location.location.contains m.0jgd | m.06n3y location.location.contains m.03qjmgc
      | m.06n3y location.location.contains m.011s2 | m.06n3y location.location.containedby m.07c51

Target:
(JOIN (R location.location.containedby) m.04zwft)
```

```
Source:
question: What does increased oxygen concentrations in the patient's lungs displace? context: Hyperbaric (
     high-pressure) medicine uses special oxygen chambers to increase the partial pressure of O 2 around
     the patient and, when needed, the medical staff. Carbon monoxide poisoning, gas gangrene, and
     decompression sickness (the 'bends') are sometimes treated using these devices. Increased O 2
     concentration in the lungs helps to displace carbon monoxide from the heme group of hemoglobin.
     Oxygen gas is poisonous to the anaerobic bacteria that cause gas gangrene, so increasing its partial
     pressure helps kill them. Decompression sickness occurs in divers who decompress too quickly after a
     dive, resulting in bubbles of inert gas, mostly nitrogen and helium, forming in their blood.
     Increasing the pressure of O 2 as soon as possible is part of the treatment.

Target:
carbon monoxide
```

```
Source:
public void delete(int key) {int i = binarySearch(mKeys, 0, mSize, key);if (i >= 0) {if (mValues[i] !=
     DELETED) {mValues[i] = DELETED;mGarbage = true;}}}

Target:
public virtual void delete(int key){int i = binarySearch(mKeys, 0, mSize, key);if (i >= 0){if (mValues[i]
     != DELETED){mValues[i] = DELETED;mGarbage = true;}}}
```

```
Source:
COc1ccc(CN(C(=O)OCc2ccccc2)[C@@H]2C(=O)N(Cc3ccc(OC)cc3OC)[C@@H]2CC=C(Br)Br)cc1

Target:
BrC(Br)(Br)Br.COc1ccc(CN(C(=O)OCc2ccccc2)[C@@H]2C(=O)N(Cc3ccc(OC)cc3OC)[C@@H]2CC=O)cc1
```

D.3 Downstream Task Specific Hyperparameters

More general fine-tuning hyperparameters and details are in Appendix B. For each task, we chose the learning rate by trying 1e-2, 3e-3, 1e-3 for Random Init finetuning, and picking the best performing value. .

CNNDM-10K We use the same input length, taget length, and max decoding length parameters as the Nonsense Summary synthetic task paper (Krishna et al., 2021).

- learning rate: 0.001
- input length: 512
- target length: 256
- max decode length: 148

MTOP

- learning rate: 0.001
- input length: 1024
- target length: 128

WebQSP

- learning rate: 0.001
- input length: 2048
- target length: 256

SQuAD

- learning rate: 0.001
- input length: 512
- target length: 128

Code Translation

- learning rate: 0.003
- input length: 1024
- target length: 1024

Retrosynthesis

- learning rate: 0.003
- input length: 256
- target length: 256

E Three Previously Proposed Synthetic Tasks from Section 3

E.1 LIME

The input token length ranges from 68 to 214 and the output token length ranges from 7 to 74. For the following LIME examples, the "rule string" is $A * A + B = C$, the "case dictionary" that represents substitutions is $\{A : a, B : b, C : d + e\}$, and the "result string" is $a * a + b = d + e$. In real generated data, the "Rule Symbols" and "Math Symbols" are randomly sampled from a vocabulary of 32K tokens for each data point.

```
Source:
<RuleSymbols> A B C <MathSymbols> * + = a b d e <s> {A : a, B : b, C : d + e} <s> a * a + b = d + e

Target:
A * A + B = C
```

```
Source:
<RuleSymbols> A B C <MathSymbols> * + = a b d e <s> A * A + B = C <s> a * a + b = d + e

Target:
{A : a, B : b, C : d + e}
```

```
Source:
<RuleSymbols> A B C <MathSymbols> * + = a b d e <s> A * A + B = C <s> {A : a, B : b, C : d + e}

Target:
a * a + b = d + e
```

E.2 Dyck Artificial Language

The input token length ranges from 3 to 512 and the output token length ranges from 3 to 114. For clarity, the following example will involve only 3 head and tail tokens represented by the symbols {, }, [,], (,). In real generated data, there will be 16K possible head-tail token pairs.

An example sentence:

```
[ ( { [ ] } [ ] ) ] { [ ] }
```

This sentence converted to a Seq2Seq task with the T5 pre-training objective:

```
Source:
[ <s0> <s1> [ ] } <s2> ] ) <s3> { [ ] <s4>

Target:
( <s0> { <s1> [ <s2> ] <s3> }
```

E.3 Nonsense Summary

The input token length ranges from 175 to 709 and the output token length ranges from 12 to 825. We provide an example of a nonsense summary data point in Appendix Figure 5, which is directly taken from the task's original paper (Krishna et al., 2021).

F Dataset Size Ablation Full Results

The table showing the full training dataset size ablation results is in Appendix Table 4.

G Simpler Synthetic Tasks from Section 4

All data is generated by sampling over 32K tokens. But for clarity, all examples are written with individual alphabetic characters representing tokens. All simpler synthetic tasks have input between length 10 and length 220.

G.1 Set

Definition Input is a token sequence. Output is the input token sequence with no duplicates (in original order).

```
Source:
a b a a b b c d

Target:
a b c d
```

Input Generation To generate one data point, we do not just uniformly sample tokens. Because over a vocabulary of 32K tokens, that simple sampling would result in few sequences with duplicate tokens where the task input and output are different. Instead, we uniformly sample the length l of the input from min length to max length, then sample t tokens that will be in the input where $t < l$, then sample the amount of each token by randomly partitioning l to t parts $\{p_1, p_2...p_t\}$, then finally generate a random input sequence based on the t sampled tokens and corresponding amounts.

G.2 Delete

Definition Input is token sequence 1 and token sequence 2. Output is sequence 1 with the first occurrence of sequence 2 deleted.

```
Source:
c b a a b b a d <s> b a

Target:
b a b b a d
```

Input Generation With 0.75 probability, generate sequence 1 and sequence 2 such that sequence 2 appears in sequence 1 at least once via the following process: sample sequence 2, sample sequence 1 length, sample location that sequence 2 appears in sequence 1, sample remaining tokens in sequence 1. With 0.25 probability, generate sequence 1 and sequence 2 such that sequence 2 does not appear in sequence 1 via the following process: sample sequence 1, sample sequence 2 and keep re-sampling sequence 2 until it does not appear in sequence 1.

G.3 Sort

Definition Input is token sequence 1 and token sequence 2. Output is input sequence 1 sorted in according to order specified by token sequence 2.

```
Source:
c b a a b c b a d <s> b a c d

Target:
b b b a a a c c d
```

Input Generation To generate token sequence 1, uniformly sample the length l of the input from min length to max length, then sample t tokens that will be in the input where $t < l$, then sample the amount of each token by randomly partitioning l to t parts $\{p_1, p_2...p_t\}$, then finally generate a random input sequence based on the t sampled tokens and corresponding amounts. To generate sequence 2, shuffle the set of tokens in sequence 1.

G.4 Identity

Definition Input is a token sequence. Output is the same as the input token sequence.

```
Source:
c b a a b

Target:
c b a a b
```

Input Generation Sample input sequence length l uniformly between min and max possible length, then sample l tokens.

G.5 Union

Definition Input is token sequence 1 and token sequence 2. Output is token sequence that is the union of the set of tokens in sequence 1 and set of tokens in sequence 2.

```
Source:
c b a a b <s> c b a e f

Target:
c b a e f
```

Input Generation Sample max length l, then sample partition of l to l_1 and l_2, which will be the lengths of sequence 1 and sequence 2. Sample $l/2$ unique tokens. To get sequence 1, sample l_1 of the unique tokens. To get sequence 2, sample l_2 of the unique tokens.

G.6 Set1 Minus Set2

Definition Input is token sequence 1 and token sequence 2. Output is a token sequence that is the subtraction of the set of tokens in sequence 2 from the set of tokens in sequence 1.

```
Source:
c b a a b <s> c a e f

Target:
b
```

Input Generation Sample max length l, then sample partition of l to l_1 and l_2, which will be the lengths of sequence 1 and sequence 2. Sample $l/2$ unique tokens. To get sequence 1, sample l_1 of the unique tokens. To get sequence 2, sample l_2 of the unique tokens.

G.7 Replace

Definition Input is token sequence 1 and token sequence 2. Token sequence 2 always has two tokens, which we will denote as t_1 and t_2. Output is token sequence 1 with t_1 replaced by t_2.

```
Source:
c b a a b <s> a x

Target:
c b x x b
```

Input Generation To get sequence 1, sample input sequence length l uniformly between min and max possible length, then sample l tokens. To get sequence 2, sample one token from the set of tokens in sequence 1, and sample a token from the entire vocab.

G.8 Duplicate

Definition Input is a token sequence. Output is the input token sequence with every token duplicated.

```
Source:
c a b a a b

Target:
c c a a b b a a a a b b
```

Input Generation Sample input sequence length l uniformly between min and max possible length, then sample l tokens.

G.9 Intersect

Definition Input is token sequence 1 and token sequence 2. Output is token sequence that is the intersection of the set of tokens in sequence 1 and set of tokens in sequence 2.

```
Source:
c a b c a a b <s> a a b

Target:
a b
```

Input Generation Sample max length l, then sample partition of l to l_1 and l_2, which will be the lengths of sequence 1 and sequence 2. Sample $l/2$ unique tokens. To get sequence 1, sample l_1 of the unique tokens. To get sequence 2, sample l_2 of the unique tokens.

G.10 Reverse

Definition Input is a token sequence. Output is the input token sequence in reverse order.

```
Source:
c a b c a a b

Target:
b a a c b a c
```

Input Generation Sample input sequence length l uniformly between min and max possible length, then sample l tokens.

G.11 Set2 Minus Set1

Definition Input is token sequence 1 and token sequence 2. Output is a token sequence that is the subtraction of the set of tokens in sequence 1 from the set of tokens in sequence 2.

```
Source:
c b c b b <s> a a b x y z

Target:
a x y z
```

Input Generation Sample max length l, then sample partition of l to l_1 and l_2, which will be the lengths of sequence 1 and sequence 2. Sample $l/2$ unique tokens. To get sequence 1, sample l_1 of the unique tokens. To get sequence 2, sample l_2 of the unique tokens.

G.12 Deduplicate

Definition Input is a token sequence. Output is the input token sequence with no adjacent duplicate tokens.

```
Source:
c b b c c d d d d e f e

Target:
c b c d e f e
```

Input Generation Uniformly sample the length l of the input from min length to max length, then sample t tokens that will be in the input where $t < l$, then sample the amount of each token by randomly partitioning l to t parts $\{p_1, p_2 ... p_t\}$, then generate the input sequence by iterating through the t tokens and repeating each one p_i times.

G.13 Last Token

Definition Input is a token sequence. Output is the last token in the input sequence.

```
Source:
c b b c c a

Target:
a
```

Input Generation Sample input sequence length l uniformly between min and max possible length, then sample l tokens.

G.14 First Token

Definition Input is a token sequence. Output is the first token in the input sequence.

```
Source:
c b b c c a

Target:
c
```

Input Generation Sample input sequence length l uniformly between min and max possible length, then sample l tokens.

G.15 Longest Word

Definition Input is a token sequence. The 'input token sequence has a special format, with special separator tokens distributed throughout. Output is the largest number of contiguous tokens.

```
Source:
c b <s> s <s> b a f <s> s d <s> a a b c w q
Target:
6
```

Input Generation Sample the number of "words" w defined as a contiguous block of normal tokens. Sample the length of the input token sequence l. Partition $l - w + 1$ to w parts $p_1, p_2, ...p_w$. Build the input sequence by sampling the "words" according to the word lengths $p_1, p_2, ...p_w$, and then joining the words with the special separator token.

G.16 Search

Definition Input is token sequence 1 and token sequence 2. Output is "yes" if token sequence 2 is contained in token sequence 1, and "no" otherwise.

```
Source:
a a b c w q a b d e <s> c w e

Target:
no
```

Input Generation With 0.75 probability, generate sequence 1 and sequence 2 such that sequence 2 appears in sequence 1 at least once by using the following procedure: sample sequence 2, sample sequence 1 length, sample location that sequence 2 appears in sequence 1, sample remaining tokens in sequence 1. With 0.25 probability, generate sequence 1 and sequence 2 such that sequence 2 does not appear in sequence 1 by using the following procedure: sample sequence 1, sample sequence 2 and keep re-sampling sequence 2 until it does not appear in sequence 1.

G.17 Length

Definition Input is a token sequence. Output is the number of tokens in the input token sequence.

```
Source:
a a b c w

Target:
5
```

Input Generation Sample input sequence length l uniformly between min and max possible length, then sample l tokens.

G.18 Count

Definition Input is token sequence 1 and token sequence 2. Token sequence 2 is only 1 token which we will denote t. Output is the number of times t occurs in token sequence 1.

Example

```
Source:
a a b c a a a w <s> a

Target:
5
```

Input Generation Start by sampling the single token t in sequence 2. Then sample l, which will be the length of sequence 1, and sample the number of times c that t will appear in token 1. Sample $l - c$ remaining tokens.

H More Initialization Schemes

We define two other initialization schemes we tried, and show results in Appendix Table 5

Across Layers Scale We tried the `Across_Layers_Scale` initialization scheme, where we initialized each *type* of parameter with the scale statistic of it computed across all layers. Some examples of types of parameters include query matrices, pre-MLP layer norms, and pre-attention layer norms. Using this initialization, all query matrices across the whole model would be initialized with a single SD, which is the SD over all the pre-trained Q matrix weights, across all layers. We find that `Across_Layers_Scale` provides significant benefits for `LIME`, `Identity`, and `Set`, preserving at least 87% of the Per Param Mean/SD initialization scheme benefit for CNNDM-10K and MTOP. This result suggests that parameter statistics varying from layer to layer might not be an important factor for synthetic pre-training initialization benefits.

Per Layer Scale We tried the `Per_Layer_Scale` initialization scheme, where every layer is initialized with just two statistics: the SD of all the non-layer norm parameter weights in that layer and the mean of all the layer norm parameter weights in that layer. Results are mixed. There is no benefit for `Identity` CNNDM-10K and minimal benefit for `LIME` MTOP. For the other four pre-training and downstream task settings, there was significant benefit, preserving at least 80% of the Per Param Mean/SD benefit for CNNDM-10K and MTOP.

I Compute Used

Each experiment was ran on a Google Cloud v3-8 TPU VM. We ran most experiments using a total of 5 such VMs.

J Additional Experiments

Multiple Seeds for Section 3 and 4 Experiments In Appendix Table 6, we show the mean and standard deviation for five seeds from evaluating Pre-trained T5, Random Init, LIME, and Set on the six downstream tasks. For LIME and Set, for each different seed, we generated and pre-trained on a new synthetic dataset.

More Tasks for Section 5.1 Per Param Mean/SD Initialization In Appendix Table 7, we show results for the Section 5.1 Per Param Mean/SD init scheme for all six tasks, not just CNNDM-10K and MTOP which were the only two tasks shown in Table 2.

Multiple Seeds for Section 5.1 Per Param Mean/SD Initialization On CNNDM-10K, MTOP, and SQuAD, we run the Per Param Mean/SD ininitialization experiments for five seeds. Results are shown in Appendix Table 7. We initialized using a model pre-trained on a different generated synthetic dataset for each seed.

More Tasks for Section 5.3 Pre-attention Layer Norm Initialization Sweep In Appendix Table 8, we show results for the Section 5.3 pre-attention layer norm initialization sweep for all six tasks, not just CNNDM-10K and MTOP which were the only two tasks shown in Table 3.

Larger Model Size In Appendix Table 9, we run experiments with T5 Base, which is a larger model size than T5 Small that was used for experiments in the main paper. T5 Base has 220 million parameters with 12 encoder/decoder layers compared to 60 million parameters with 6 encoder/decoder layers for T5 Small. `LIME`, `Set`, `LIME Per_Param_Mean_SD`, and `Set Per_Param_Mean_SD` close a large proportion of the gap to natural pre-training, similar to results for the T5 Small experiments.

Table 4: Dataset size ablation. Retrosynthesis-40K and SQuAD-87K are the same as Retrosynthesis and SQuAD in Table 1. The CNNDM-10K and CNNDM-100K results as well as the Retrosynthesis-10K and Reotrosynthesis-40K results show that benefits from synthetic (and T5 natural language) pre-training increase as the amount of training data for a task decrease.

	Retrosyn-1K	Retrosyn-10K	Retrosyn-40K
Pre-trained T5	7.6	29.5	43.1
LIME	4.0	25.1	41.1
Random Init	0.0	18	39.2

	CNNDM-1K	CNNDM-10K	CNNDM-100K
Pre-trained T5	29.6	35.8	39.1
LIME	28.4	33.2	37.6
Random Init	12.2	18.9	35.1

	SQuAD-1K	SQuAD-10K	SQuAD-87K
Pre-trained T5	19.5/36.8	59.2/73.5	77.5/86.0
LIME	3.7/8.5	23.21/34.1	50.4/62.1
Random Init	0.8/0.9	0.3/1.6	17.2/25.2

Table 5: Evaluating different initialization schemes with statistics of LIME, Set, and Identity pre-trained models.

	CNNDM-10K			MTOP		
	LIME	Set	Identity	LIME	Set	Identity
Section 5.1						
Full Init	33.2	32.8	30.2	73.7/94.0	71.7/93.7	68.6/93.0
Per Param Mean/SD	29.3	29.8	28.8	66.1/92.3	66.6/92.3	67.8/92.6
Section 5.2						
Per Param Scale	29.0	29.5	29.0	67.3/92.5	66.5/92.4	66.3/92.3
Across Layers Scale	27.8	28.8	28.5	63.5/91.5	66.9/92.5	67.5/92.6
Per Layer Scale	27.1	28.1	19.6	48.1/85.3	61.1/90.5	60.9/90.5
Whole Model Scale	15.5	26.1	20.5	34.3/81.4	38.7/83.1	59.8/91.0
Pre-attn LN Per Param Scale	27.8	25.1	27.3	59.9/90.3	58.9/89.0	46.7/86.3
Attention Per Param Scale	17.2	19.1	19.1	19.3/73.9	41.5/84.9	38.4/82.9
Pre-MLP LN Per Param Scale	20.2	20.6	20.5	41.8/85.1	40.3/84.5	43.0/85.4
MLP Per Param Scale	16.6	18.7	18.3	36.5/80.8	36.5/81.6	37.9/83.0

Baseline	CNNDM-10K	MTOP
Random Init	18.9	38.6/83.0
Pre-trained T5	35.8	81.0/95.2

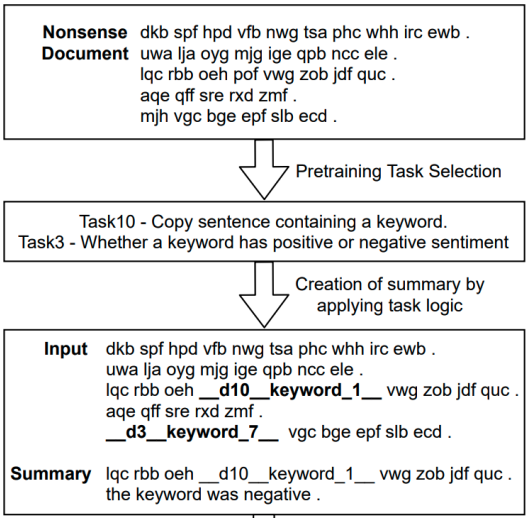

Figure 5: Example of a Nonsense Summary task data point. Figure is directly taken from (Krishna et al., 2021).

Table 6: Same table as Table 1 in main paper except with mean and standard deviation over 5 seeds for Pre-trained T5, Random Init, LIME, and Set. For LIME and Set, for each seed, a different synthetic pre-training set was generated and pre-trained on.

	CNNDM-10K ROUGE1	MTOP EM/F1	WebQSP EM/F1	SQuAD EM/F1	Code Trans. EM	Retrosyn. EM
Baselines						
Pre-trained T5	**35.8 (0.1)**	**81.0/95.2 (0.3/2.2)**	**82.3/91.7 (0.8/0.4)**	**77.4/86.1 (0.2/0.2)**	**61.2 (0.7)**	**43.5 (0.3)**
Wiki 10K	34.0	71.6/94.2	79.6/90.6	67.1/76.9	60.2	41.1
Random Init	18.9 (0.5)	38.6/83.0 (2.1/1.2)	25.8/71.5 (0.7/0.5)	16.0/24.2 (2.1/2.0)	57.6 (0.3)	39.0 (0.5)
Section 3.1: Previously Proposed Synthetic Tasks						
LIME	**33.2 (0.7)**	**73.7/94.0 (0.8/0.1)**	**76.2/89.6 (1.2/0.6)**	**49.6/61.1 (1.6/1.3)**	**58.3 (1.1)**	**40.7 (0.4)**
Dyck	27.1	65.9/91.9	58.5/83.5	50.3/62.4	58.8	40.4
Nons. Summary	32.0	68.0/92.7	65.2/85.2	48.4/60.1	57.3	39.6
Section 4.1: Simpler Synthetic Tasks						
Set	**32.8 (0.5)**	**71.7/93.7 (0.9/0.2)**	**73.6/88.6 (1.0/0.3)**	**48.2/60.0 (0.5/0.7)**	**58.8 (0.8)**	**40.5 (0.9)**
Identity	30.2	68.6/93.0	69.8/86.8	26.1/35.4	57.8	40.5

Table 7: Section 5.1 Per Param Mean/SD initialization results for all downstream tasks (rather than just CNNDM-10K and MTOP which were the only two shown in Table 2). Baseline results and normal fune-tune results are from Appendix Table 6. For CNNDM-10K, MTOP, and SQuAD we provide the mean/SD for five seeds, where for each seed we ran the initialization with a different pre-trained model with different generated data. Per Param Mean/SD initialization gave CNNDM-10K, MTOP, SQuAD, and WebQSP noticeable gains over Random Init, close to gains from Full Init synthetic pre-trained models.

	CNNDM-10K			MTOP		
	LIME	Set	Identity	LIME	Set	Identity
Full Init	33.2 (0.7)	32.8 (0.5)	30.2	73.7/94.0 (0.8/0.1)	71.7/93.7 (0.9/0.2)	68.6/93.0
Per Param Mean/SD	28.7 (0.7)	29.4 (0.6)	29.2 (0.6)	67.9/92.6 (3.3/0.8)	67.3/92.7 (2.5/0.6)	66.6/92.2 (1.4/0.5)

Baseline	CNNDM-10K	MTOP
Random Init	18.9 (0.5)	38.6/83.0 (2.1/1.2)
Pre-trained T5	35.8 (0.1)	81.0/95.2 (0.3/2.2)

	SQuAD			WebQSP		
	LIME	Set	Identity	LIME	Set	Identity
Full Init	49.6/61.1 (1.6/1.3)	48.2/60.0 (0.5/0.7)	26.1/35.4	76.2/89.6 (1.2/0.6)	73.6/88.6 (1.0/0.3)	69.8/86.8
Per Param Mean/SD	35.4/45.5 (9.2/10.4)	31.1/40.4 (6.5/7.4)	25.6/34.4 (4.9/5.7)	60.9/84.8	63.0/86.1	58.9/83.6

Baseline	SQuAD	WebQSP
Random Init	16.0/24.2 (2.1/2.0)	25.8/71.5 (0.7/0.5)
Pre-trained T5	77.4/86.1 (0.2/0.2)	82.3/91.7 (0.8/0.4)

	Code Trans.			Retrosyn.		
	LIME	Set	Identity	LIME	Set	Identity
Full Init	58.3 (1.1)	58.8 (0.8)	57.8	40.7 (0.4)	40.5 (0.9)	40.5
Per Param Mean/SD	58.0	58.4	57.2	40.3	38.7	39.3

Baseline	Code Trans.	Retrosyn.
Random Init	57.6 (0.3)	39.0 (0.5)
Pre-trained T5	61.2 (0.7)	43.5 (0.3)

Table 8: Pre-attention layer norm initialization sweep results for all downstream tasks (rather than just CNNDM-10K and MTOP which were the only two shown in Table 3). Baseline results are from Appendix Table 6. Lower pre-attention layer norm value gave noticeable benefits over Random Init for CNNDM-10K, MTOP, WebQSP, and SQuAD.

Pre-attn LN Init Value	CNNDM-10K	MTOP	WebQSP	SQuAD	Code Trans.	Retrosyn.
0.05	28.2	25.7/73.3	30.5/71.2	22.8/31.2	57.9	39.3
0.1	28.3	32.5/77.5	28.9/70.6	23.5/31.7	57.6	39.9
0.2	28.2	56.4/89.3	30.6/72.3	28.2/37.3	57.0	39.4
0.4	24.4	56.9/89.9	30.0/72.8	36.3/46.9	57.3	39.6
0.8	18.6	50.7/88.5	28.0/71.6	34.2/44.6	57.2	39.9

Baseline	CNNDM-10K	MTOP	WebQSP	SQuAD	Code Trans.	Retrosyn.
Random Init	18.9 (0.5)	38.6/83.0 (2.1/1.2)	25.8/71.5 (0.7/0.5)	16.0/24.2 (2.1/2.0)	57.6 (0.3)	39.0 (0.5)
Pre-trained T5	35.8 (0.1)	81.0/95.2 (0.3/2.2)	82.3/91.7 (0.8/0.4)	77.4/86.1 (0.2/0.2)	61.2 (0.7)	43.5 (0.3)

K T5 Small Parameters List

```
decoder/decoder_norm/scale
decoder/layers_0/encoder_decoder_attention/key/kernel
decoder/layers_0/encoder_decoder_attention/out/kernel
decoder/layers_0/encoder_decoder_attention/query/kernel
decoder/layers_0/encoder_decoder_attention/value/kernel
decoder/layers_0/mlp/wi/kernel
decoder/layers_0/mlp/wo/kernel
decoder/layers_0/pre_cross_attention_layer_norm/scale
decoder/layers_0/pre_mlp_layer_norm/scale
decoder/layers_0/pre_self_attention_layer_norm/scale
decoder/layers_0/self_attention/key/kernel
decoder/layers_0/self_attention/out/kernel
decoder/layers_0/self_attention/query/kernel
decoder/layers_0/self_attention/value/kernel
decoder/layers_1/encoder_decoder_attention/key/kernel
decoder/layers_1/encoder_decoder_attention/out/kernel
decoder/layers_1/encoder_decoder_attention/query/kernel
decoder/layers_1/encoder_decoder_attention/value/kernel
decoder/layers_1/mlp/wi/kernel
decoder/layers_1/mlp/wo/kernel
decoder/layers_1/pre_cross_attention_layer_norm/scale
decoder/layers_1/pre_mlp_layer_norm/scale
decoder/layers_1/pre_self_attention_layer_norm/scale
decoder/layers_1/self_attention/key/kernel
decoder/layers_1/self_attention/out/kernel
decoder/layers_1/self_attention/query/kernel
decoder/layers_1/self_attention/value/kernel
decoder/layers_2/encoder_decoder_attention/key/kernel
decoder/layers_2/encoder_decoder_attention/out/kernel
decoder/layers_2/encoder_decoder_attention/query/kernel
decoder/layers_2/encoder_decoder_attention/value/kernel
decoder/layers_2/mlp/wi/kernel
decoder/layers_2/mlp/wo/kernel
decoder/layers_2/pre_cross_attention_layer_norm/scale
decoder/layers_2/pre_mlp_layer_norm/scale
decoder/layers_2/pre_self_attention_layer_norm/scale
decoder/layers_2/self_attention/key/kernel
decoder/layers_2/self_attention/out/kernel
decoder/layers_2/self_attention/query/kernel
decoder/layers_2/self_attention/value/kernel
decoder/layers_3/encoder_decoder_attention/key/kernel
decoder/layers_3/encoder_decoder_attention/out/kernel
decoder/layers_3/encoder_decoder_attention/query/kernel
decoder/layers_3/encoder_decoder_attention/value/kernel
decoder/layers_3/mlp/wi/kernel
decoder/layers_3/mlp/wo/kernel
decoder/layers_3/pre_cross_attention_layer_norm/scale
decoder/layers_3/pre_mlp_layer_norm/scale
decoder/layers_3/pre_self_attention_layer_norm/scale
decoder/layers_3/self_attention/key/kernel
decoder/layers_3/self_attention/out/kernel
decoder/layers_3/self_attention/query/kernel
decoder/layers_3/self_attention/value/kernel
decoder/layers_4/encoder_decoder_attention/key/kernel
decoder/layers_4/encoder_decoder_attention/out/kernel
decoder/layers_4/encoder_decoder_attention/query/kernel
decoder/layers_4/encoder_decoder_attention/value/kernel
decoder/layers_4/mlp/wi/kernel
decoder/layers_4/mlp/wo/kernel
decoder/layers_4/pre_cross_attention_layer_norm/scale
decoder/layers_4/pre_mlp_layer_norm/scale
decoder/layers_4/pre_self_attention_layer_norm/scale
decoder/layers_4/self_attention/key/kernel
decoder/layers_4/self_attention/out/kernel
decoder/layers_4/self_attention/query/kernel
decoder/layers_4/self_attention/value/kernel
decoder/layers_5/encoder_decoder_attention/key/kernel
decoder/layers_5/encoder_decoder_attention/out/kernel
decoder/layers_5/encoder_decoder_attention/query/kernel
decoder/layers_5/encoder_decoder_attention/value/kernel
decoder/layers_5/mlp/wi/kernel
decoder/layers_5/mlp/wo/kernel
decoder/layers_5/pre_cross_attention_layer_norm/scale
decoder/layers_5/pre_mlp_layer_norm/scale
decoder/layers_5/pre_self_attention_layer_norm/scale
decoder/layers_5/self_attention/key/kernel
decoder/layers_5/self_attention/out/kernel
decoder/layers_5/self_attention/query/kernel
```

```
decoder/layers_5/self_attention/value/kernel
decoder/relpos_bias/rel_embedding
encoder/encoder_norm/scale
encoder/layers_0/attention/key/kernel
encoder/layers_0/attention/out/kernel
encoder/layers_0/attention/query/kernel
encoder/layers_0/attention/value/kernel
encoder/layers_0/mlp/wi/kernel
encoder/layers_0/mlp/wo/kernel
encoder/layers_0/pre_attention_layer_norm/scale
encoder/layers_0/pre_mlp_layer_norm/scale
encoder/layers_1/attention/key/kernel
encoder/layers_1/attention/out/kernel
encoder/layers_1/attention/query/kernel
encoder/layers_1/attention/value/kernel
encoder/layers_1/mlp/wi/kernel
encoder/layers_1/mlp/wo/kernel
encoder/layers_1/pre_attention_layer_norm/scale
encoder/layers_1/pre_mlp_layer_norm/scale
encoder/layers_2/attention/key/kernel
encoder/layers_2/attention/out/kernel
encoder/layers_2/attention/query/kernel
encoder/layers_2/attention/value/kernel
encoder/layers_2/mlp/wi/kernel
encoder/layers_2/mlp/wo/kernel
encoder/layers_2/pre_attention_layer_norm/scale
encoder/layers_2/pre_mlp_layer_norm/scale
encoder/layers_3/attention/key/kernel
encoder/layers_3/attention/out/kernel
encoder/layers_3/attention/query/kernel
encoder/layers_3/attention/value/kernel
encoder/layers_3/mlp/wi/kernel
encoder/layers_3/mlp/wo/kernel
encoder/layers_3/pre_attention_layer_norm/scale
encoder/layers_3/pre_mlp_layer_norm/scale
encoder/layers_4/attention/key/kernel
encoder/layers_4/attention/out/kernel
encoder/layers_4/attention/query/kernel
encoder/layers_4/attention/value/kernel
encoder/layers_4/mlp/wi/kernel
encoder/layers_4/mlp/wo/kernel
encoder/layers_4/pre_attention_layer_norm/scale
encoder/layers_4/pre_mlp_layer_norm/scale
encoder/layers_5/attention/key/kernel
encoder/layers_5/attention/out/kernel
encoder/layers_5/attention/query/kernel
encoder/layers_5/attention/value/kernel
encoder/layers_5/mlp/wi/kernel
encoder/layers_5/mlp/wo/kernel
encoder/layers_5/pre_attention_layer_norm/scale
encoder/layers_5/pre_mlp_layer_norm/scale
encoder/relpos_bias/rel_embedding
```

Table 9: We run experiments with T5 Base, which is a larger model size than T5 Small that was used for experiments in the main paper. T5 Base has 220 million parameters with 12 encoder/decoder layers compared to 60 million parameters with 6 encoder/decoder layers for T5 Small. `LIME`, `Set`, `LIME Per_Param_Mean_SD`, and `Set Per_Param_Mean_SD` close a large proportion of the gap to natural pre-training, similar to results for the T5 Small experiments.

Experiments With Larger Model Size (T5 Base)

	CNNDM-10K ROUGE 1	MTOP EM/F1	SQuAD EM/F1	Retrosynthesis EM	Average	
Pre-trained_T5	37.6	83.8/96.8	83.6/91.1	44.4	62.4	100.0%
Random_Init	16.4	30.30/78.3	7.5/9.9	36.9	22.8	0.0%
LIME	33.7	72.5/93.8	48.4/60.4	38.2	48.2	57.9%
Set	34.4	74.4/94.2	51.8/62.7	39.6	50.1	65.4%
LIME Per_Param_Mean_SD	26.5	59.2/90.5	25.9/34.2	37.3	37.2	32.8%
Set Per_Param_Mean_SD	27.4	64.9/92.2	22.4/30.6	37.5	38.1	36.0%

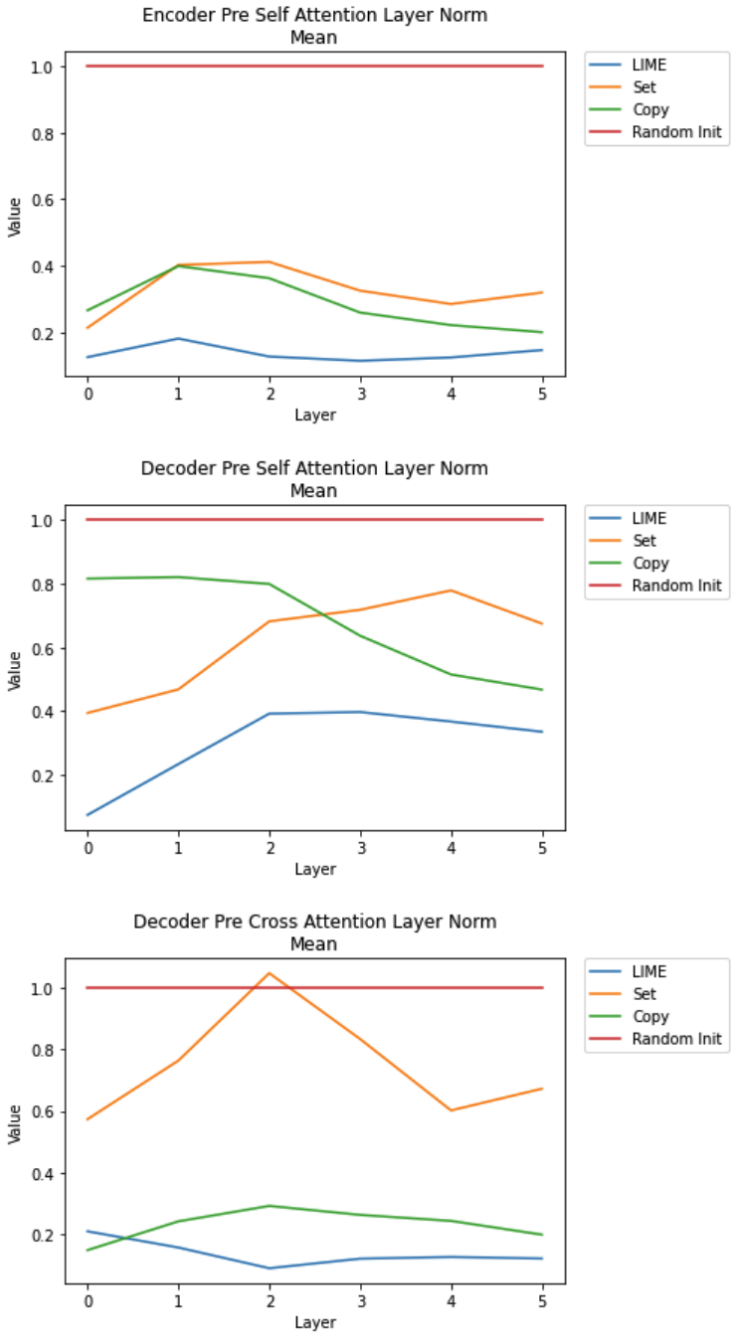

Figure 6: For each layer, compute the mean over weights of the pre-attention layer norm parameter in that layer. Compute and plot these values for a Random Init model as well as LIME, Set, and Identity pre-trained models. For example, looking at the top-most plot, the mean of the Set pre-trained model's encoder layer 0 pre-attention layer norm is about 0.2. And the mean of the Identity pre-trained model's encoder layer 1 pre-attention layer norm is about 0.4. Looking at the bottom-most plot, the mean of the Set pre-trained model's encoder layer 4 pre-cross-attention layer norm is about 0.6.