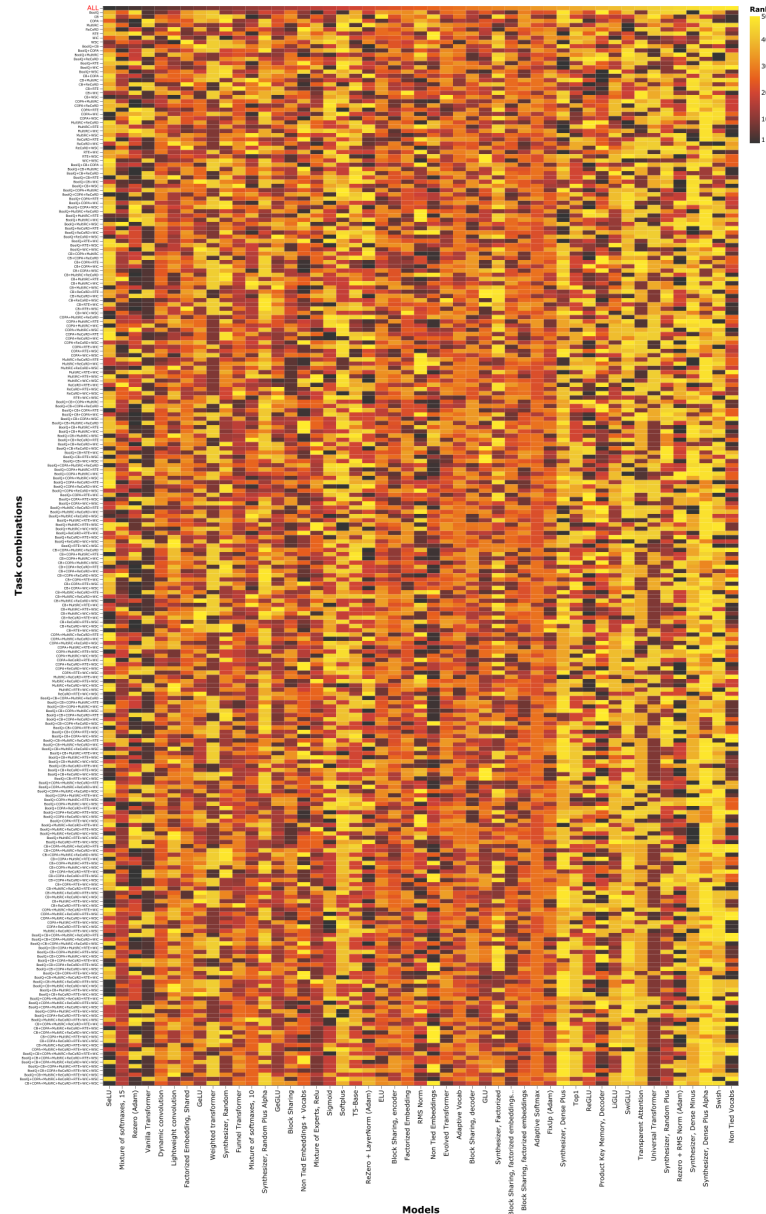


## 933

## 934

## 935

936  
937  
938  
939



940

## A.2 SuperGLUE: Rank correlation between SuperGLUE score and individual tasks

Figure 4 shows the rank correlation between the SuperGLUE score and each of the 8 tasks in the benchmark, given 55 different models described in [Narang et al., 2021]. The average Kendall rank correlation of tasks with the SuperGLUE score is 0.648. This correlation is not perfect, but a more important point in the SuperGLUE benchmark is the disagreement of the top-k models across all tasks. This point is highlighted in Figure 1, where for instance, in 6 out of 8 individual tasks, we have different models as the winner. Thus using the mean score for a practitioner to choose a model to adapt it for their own application can be sub-optimal based on the context.

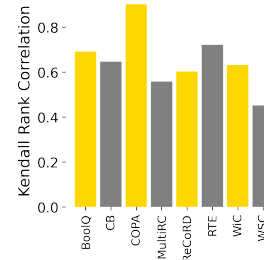


Figure 4: Rank correlation between the SuperGLUE mean score and task’s scores.

## A.3 SuperGLUE: Effect on relative ranking of models

A significant amount of work in machine learning modeling is to determine to the relative performance of a set of different inductive biases or model architectures. We show that the ranking of models can be drastically altered based on the choice of the subset of the benchmark considered. In other words, the relative performance of models can be easily manipulated by task selection. In order to show this phenomenon, we select ten models, namely vanilla Transformers, Weighted Transformers, Funnel Transformers, Switch Transformers, Lightweight Convolutions, Dynamic Convolutions, Universal Transformers and Adaptive Softmax. The results are similarly obtained from [Narang et al., 2021]. Table 1 reports a sample of different selection of tasks. We show that for a different selection of tasks, the relative order of model performance is *very* different. Notably, models such as Universal, MoE, Adaptive Softmax, Switch all take turns to become the best performing model on multiple task configurations. Consequently, it is easy to see that with some manipulation of the benchmark configuration, it is easy to endorse and favor the performance of one model over another.

Table 1: Relative order of different models when selecting different subsets of SuperGLUE. Selecting different subsets of tasks can produce *very different* outcomes for relative ranking of model architectures. Models that did not appear in Top-5 at all are Lightweight Conv, Dynamic Conv and Transparent Attention. For tasks, A=BoolQ, B=CB, C=CoPA, D=MultiRC, E=ReCoRD, F=RTE, G=WiC, H=WSC.

Tasks	Top-5 Performing Models (In Order)
H	Universal, Switch, Adaptive Softmax, Weighted, Vanilla
G	MoE, Switch, Vanilla, Funnel, Universal
A, B	Adaptive Softmax, Vanilla, MoE, Switch, Weighted
A, C	MoE, Switch, Adaptive Softmax, Vanilla, Universal
D, H	Switch, Universal, Adaptive Softmax, MoE, Weighted
B, E, H	Adaptive Softmax, Switch, MoE, Vanilla, Weighted
F, G, H	Switch, MoE, Adaptive Softmax, Universal, Vanilla
A, F, G	MoE, Switch, Vanilla, Adaptive Softmax, Vanilla
C, F, G, H	Switch, MoE, Adaptive Softmax, Vanilla, Universal
A, C, D, G	MoE, Switch, Adaptive Softmax, Vanilla, Universal
All	Switch, MoE, Adaptive Softmax, Vanilla, Universal

Optimistically, we also note that even under the notion of a *lottery*, not all models have equal odds. Models that perform poorly across all tasks generally tend to not have a chance to qualify for the Top-5 of any of the above benchmark configurations. We note that Lightweight convolutions, Dynamic Convolutions, and Transparent Attention never made it to any of the Top-5 rankings. Models such as Funnel and Weighted also make very limited appearances. In short, we show empirically that benchmark suites can do pretty well in filtering model architectures that do poorly on most tasks.

#### A.4 VTAB: Details about the tasks, categories, and models.

VTAB is used for evaluating the quality of representations learned by different models in terms of their ability to adapt to diverse, unseen tasks with few examples. In addition to standard natural image tasks, like classification on ImageNet or CIFAR datasets, VTAB includes tasks that are related to sensorimotor control, medical imaging and scene understanding. The benchmark defines the score of algorithm as its expected performance over a known distribution of tasks that includes those that a human can solve, from visual input alone.

VTAB defines a total of 19 tasks, grouped into three categories: (i) *Natural*, which contains natural images captured using standard cameras that represent generic, fine-grained, or abstract objects [Caltech101 [Fei-Fei et al., 2006], CIFAR100 [Krizhevsky et al., 2009], DTD [Cimpoi et al., 2014], Flowers102 [Nilsback and Zisserman, 2008], Pets [Parkhi et al., 2012], Sun397 [Xiao et al., 2010], and SVHN [Netzer et al., 2011].]; (ii) *Specialized*, which contains images of the world that captured through specialist equipment [Remote sensing: Resisc45 [Cheng et al., 2017] and EuroSAT [Helber et al., 2019]: aerial images of the Earth captured using satellites or aerial photography; Medical: Patch Camelyon [Veeling et al., 2018], metastases detection from microscopy images, and Diabetic Retinopathy [Kaggle and EyePacs, 2015], retinopathy classification from fundus images.]; and finally (iii) *Structured*, which contains tasks that designed to assess comprehension of the structure of a scene, mostly generated syntactically using simulated environments [CLEVR [Johnson et al., 2017]: Simple shapes rendered in a 3D scene, with two tasks: counting and depth prediction, dSprites [Higgins et al., 2016]: Simple black-and-white shapes rendered in 2D, with two tasks: location and orientation prediction, SmallNORB [LeCun et al., 2004]: Artificial objects viewed under varying conditions, with two tasks: object azimuth and camera-elevation prediction, DMLab [Beattie et al., 2016]: Frames from a rendered 3D maze. The task involves predicting the time for a pre-trained RL agent to navigate to an object, KITTI [Geiger et al., 2013]: frames captured from a car driver’s perspective and the task is to predict the depth of the nearest vehicle.]. We have evaluated 32 different models against all the 19 VTAB tasks. The difference between models is on their architectures (e.g. WAE-GAN [Tolstikhin et al., 2017] vs. VIVI [Tschannen et al., 2020]), their sizes (e.g. ResNet-50 vs. ResNet-101 [Kolesnikov et al., 2019]), or the dataset they were pre-trained on (e.g. ResNet-50 pretrained on ImageNet-21k vs. ResNet-50 pretrained on JFT [Kolesnikov et al., 2019]). Models we considered in our study are those that are introduced as “representation learning algorithms” in [Zhai et al., 2019].

#### B VTAB: Agreement on top-ranked models across sub-categories and tasks

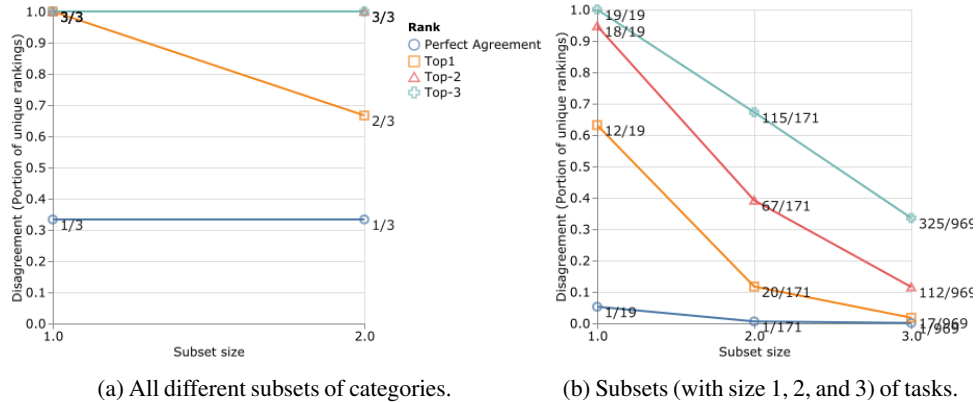


Figure 5: Disagreement of model rankings on the VTAB benchmark as a function of the number of selected benchmark sub-categories (3 sub-categories: Natural, Specialized, Structured) or tasks (19 different tasks).

Similar to Figure 1, we looked into the disagreement of Top-1, 2, and 3 models based on different combinations of three VTAB sub-categories as well as different combinations with sizes 1, 2, and 3 of VTAB tasks. The  $x$ -axis represents the number of sub-categories/tasks in each sub-selection and each line corresponds to a different value of  $k$  for the Top- $k$  in the rankings. Points are labels as  $A/B$ , where  $A$  is the number of unique model rankings and  $B$  is the total number of possible sub-category combinations for this subset size.

In Figure 5a, we can see that all categories disagree on the winning model (top-1) and there is no full agreement on the set of top-2 or top-3 models. We can see a similar disagreement between subsets of

tasks in Figure 5b. For instance, out of 19 individual tasks (the subset of size 1), there are 12 different winners (top-1) model or looking at the subsets of size 2 tasks, there are 20 different winners. Note that although the disagreement portion is 20/171 is rather small, at the end of the day, we have 20 different models performing best in different situations, and taking a single model based on the VTAB score as the best one can be easily a sub-optimal choice for many scenarios.

### B.0.1 Long Range Arena

Table 2: Top 3 performing models on LRA depending on which subset of tasks we select.

Task	Best Model	Rank-2	Rank-3
$t_1$ (Text only)	Linear Transformers	Performer	Transformer
$t_2$ (Retrieval only)	Sparse Transformers	BigBird	Longformer
$t_3$ (ListOps only)	Reformer	Synthesizer	Transformer
$t_4$ (Image only)	Sparse Transformer	Performer	Transformer
$t_5$ (Path only)	Performer	Linformer	Linear Transformers
$t_1 + t_2$	BigBird	Sparse Transformer	Transformer
$t_1 + t_3$	Transformer	BigBird	Synthesizer
$t_1 + t_4$	Linear Transformer	Performer	Transformer
$t_1 + t_5$	Performer	BigBird	Transformer
$t_2 + t_3$	BigBird	Transformer	Longformer
$t_2 + t_4$	Sparse Transformer	BigBird	Transformer
$t_2 + t_5$	BigBird	Sparse Transformer	Performer
$t_3 + t_5$	Linformer	BigBird	Transformer
$t_3 + t_4$	Transformer	Synthesizer	Longformer
$t_4 + t_5$	Performer	Linear Transformer	Sparse Transformer
$t_1 + t_2 + t_3$	BigBird	Transformer	Synthesizer
$t_1 + t_2 + t_4$	Sparse Transformer	Transformer	BigBird
$t_1 + t_2 + t_5$	Performer	Linear Transformer	Transformer
$t_2 + t_3 + t_4$	Transformer	Longformer	Synthesizer
$t_2 + t_3 + t_5$	BigBird	Transformer	Longformer
$t_3 + t_4 + t_5$	BigBird	Transformer	Longformer
$t_1 + t_2 + t_3 + t_4$	Transformer	BigBird	Longformer
$t_1 + t_3 + t_4 + t_5$	BigBird	Transformer	Longformer
$t_1 + t_2 + t_4 + t_5$	Sparse Transformer	Performer	BigBird
$t_2 + t_3 + t_4 + t_5$	BigBird	Transformer	Longformer
$t_1 + t_2 + t_3 + t_4 + t_5$ (LRA Score)	BigBird	Transformer	Longformer

The Long Range Arena (LRA; Tay et al. [2020b]) is a benchmark designed for aggregated evaluation of long-range Transformer models [Tay et al., 2020c]. Similar to other benchmark suites, LRA consists of six tasks: ListOps, Long Text Classification, Long Text Retrieval, Pixel-wise Image Classification, and two variants of spatial reasoning based on the path-finder task. The authors rank eleven efficient transformer models by aggregating performance across all six tasks. To demonstrate that here too task selection matters, we computed Top-3 rankings of models for each task combination displayed in Table 2. Model name abbreviations are used for brevity and because the actual model names are not important for the purpose of this analysis. Notably, it is easy to see that the identity of each of the top-3 changes frequently as the subset of evaluation tasks is changed.

### B.0.2 RL Unplugged

RL Unplugged [Gulcehre et al., 2020] is a suite of benchmarks for offline reinforcement learning, where the task for the agent is to learn a policy directly from some logged data that is produced by a system as part of its normal operation, without interacting with the environment at the time of learning. In reinforcement learning, in general, it has been shown that varying random seeds alone can lead to a high variance between runs [Henderson et al., 2018], and this seed lottery is introducing difficulty in comparing different methods and making conclusions. Here, we study offline RL, where the results are more stable for the sake of focusing a bit more on the task selection bias problem. We will discuss online RL and expand on some other aspects in the context of Section D.1. RL Unplugged introduces a collection of task domains and associated datasets together with a clear evaluation protocol. It includes some widely used domains such as the DM Control Suite [Tassa et al., 2018] and Atari 2600 games [Bellemare et al., 2013], as well as Real-World RL (RWRL) tasks [Dulac-Arnold et al., 2019] and DM Locomotion tasks [Heess et al., 2017].

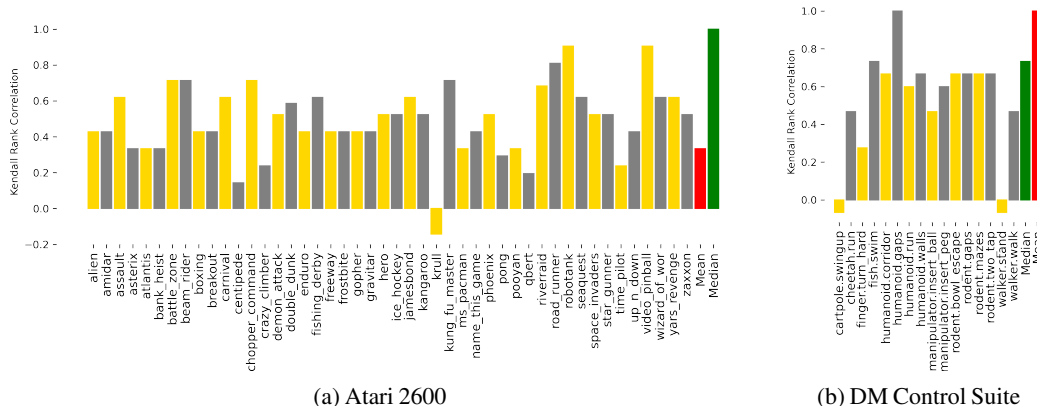


Figure 6: Rank correlation between the aggregated score and scores from each individual dataset. Note that the common approach in the literature to ingrate scores is “*median* human normalized performance” for Atari and “*mean*” for DM controls.

Here, we study the aggregated scores over multiple tasks in Atari 2600 and DM Control from RL-Unplugged. Atari 2600 consists of 46 Atari games, and DM Control has 9 diffident tasks. We use the performance of 7 differed baselines<sup>14</sup> in our analysis.

Figure 6a presents the Kendall rank correlation when ranking different models based on their human normalized performance on each task vs the *median* human normalized performance across all tasks. We also show the correlation between median and mean human normalized performance on Atari. Although many papers reported mean performance on Atari as the aggregated score, it is becoming a standard to report median since the mean is potentially less informative, as it is dominated by a few games (e.g. Atlantis) where agents achieve scores orders of magnitude higher than humans do. Figure 6b also shows the Kendall rank correlation of the mean performance across all tasks with performance on each task as well as the median. First of all, in both cases, it can be seen that the ranking of models based on individual tasks can widely disagree the ranking from the aggregated score (average rank correlation in Figure 6a is  $\approx 0.49$  and in Figure 6b is  $\approx 0.54$ ), indicating how solely reporting the aggregated score can send a potentially wrong signal for choosing the best model. Moreover, the aggregation strategies, i.e. mean and median in this case do not agree which shows standardizing one over another with the intention of considering only one of them comes at the cost of losing some information.

### B.1 Example: GLUE benchmark

The GLUE benchmark was pitched as a general language understanding benchmark and is an aggregation of 8 datasets that have been previously proposed [Williams et al., 2017]. We use this as an example of a community bias. To this date, the majority of pretrained LM paper evaluates on the GLUE benchmark. This includes widely recognized and cited papers such as BERT [Devlin et al., 2018], ALBERT [Lan et al., 2019], RoBERTa [Liu et al., 2019], XLNet [Yang et al., 2020], ELECTRA, and many others.<sup>15</sup> Given the popularity of GLUE for evaluating these models, it is only imperative that newly proposed models have to shine on GLUE in order to increase their likelihood of acceptance (in the literal sense or by the community).

Here, it is good to note that seven out of eight tasks in GLUE are actually *matching* tasks that require modeling the relationship between two or more sequences. While it is still unclear how much this problem formulation has to do with natural language understanding, it is clear that this problem formulation favors a certain class of models (e.g., Transformers which has this baked-in cross attention in the encoder). It is easy to see that this conflates an actual advantage in problem formulation (and input setting) with the ability of an encoder model to learn textual representations. While one may argue that a method should reap rewards even for a problem formulation advantage, it is also good to note that many of these cross attention setups are infeasible in practice at scale [Guo et al., 2019, Seo et al., 2018]. It is also interesting that, if the tasks in GLUE were swapped for other equally plausible

<sup>14</sup>For our analysis we used the data from the ancillary files of [Schrittwieser et al., 2021], which can be found in <https://arxiv.org/src/2104.06294v1/anc>.

<sup>15</sup>We have manually checked the papers presented ideas to improve pretrained LMs with more than 500 citations that and in all these papers GLUE has been used for evaluation.

1074 and practical tasks, we might encourage the development of alternative architectures such as pretrained  
1075 ConvNets in NLP [Tay et al., 2021].

## 1076 C Community bias example

1077 The GLUE benchmark was pitched as a general language understanding benchmark and is an aggre-  
1078 gation of 8 datasets that have been previously proposed [Williams et al., 2017]. We use this as an  
1079 example of a community bias. To this date, the majority of pretrained LM paper evaluates on the GLUE  
1080 benchmark. This includes widely recognized and cited papers such as BERT [Devlin et al., 2018],  
1081 ALBERT [Lan et al., 2019], RoBERTa [Liu et al., 2019], XLNet [Yang et al., 2020], ELECTRA, and  
1082 many others.<sup>16</sup> Given the popularity of GLUE for evaluating these models, it is only imperative that  
1083 newly proposed models have to shine on GLUE in order to increase their likelihood of acceptance (in  
1084 the literal sense or by the community).

1085 Here, it is good to note that seven out of eight tasks in GLUE are actually *matching* tasks that require  
1086 modeling the relationship between two or more sequences. While it is still unclear how much this  
1087 problem formulation has to do with natural language understanding, it is clear that this problem  
1088 formulation favors a certain class of models (e.g., Transformers which has this baked-in cross attention  
1089 in the encoder). It is easy to see that this conflates an actual advantage in problem formulation (and  
1090 input setting) with the ability of an encoder model to learn textual representations. While one may  
1091 argue that a method should reap rewards even for a problem formulation advantage, it is also good to  
1092 note that many of these cross attention setups are infeasible in practice at scale [Guo et al., 2019, Seo  
1093 et al., 2018]. It is also interesting that, if the tasks in GLUE were swapped for other equally plausible  
1094 and practical tasks, we might encourage the development of alternative architectures such as pretrained  
1095 ConvNets in NLP [Tay et al., 2021].

## 1096 D Rigging the lottery: additional case study

### 1097 D.1 ALE and evaluation setup inconsistencies

1098 An example of a benchmark that hundreds of papers have used as a testbed, while simultaneously  
1099 employing a number of distinct experimental evaluation protocols is the Arcade Learning Environment  
1100 (ALE) which is based on Atari 2600 games [Mnih et al., 2013]. The main aspects in which evaluation  
1101 setups in different papers using ALE diverge are different metrics used for summarizing agent perfor-  
1102 mance, and the different mechanisms used for injecting stochasticity in the environment [Machado  
1103 et al., 2018].

1104 For example, different assumptions can be made for determining episode termination. While in some  
1105 publications episodes terminate when the game is over [Bellemare et al., 2013, Hausknecht et al., 2014,  
1106 Liang et al., 2015, Lipovetzky et al., 2015, Martin et al., 2017], while others papers choose to terminate  
1107 the training episodes for a subset of the games when the agent loses a life [Mnih et al., 2016, Nair et al.,  
1108 2015, Wang et al., 2016, Van Hasselt et al., 2016].

1109 Another major disagreement in evaluation strategies for ALE, also comes from using different param-  
1110 eters used for the evaluation setup. For example, some papers use a non-default value for the skipframe  
1111 parameter<sup>17</sup> in their baseline models [Mnih et al., 2015]. Alternatively in some publication, methods are  
1112 evaluated for each  $2 \times 10^5$  frames [Pritzel et al., 2017], while in others methods are evaluated every  $10^6$   
1113 frames [Mnih et al., 2013, 2016]. Another observation is the difference between the number of games  
1114 used in the evaluation setups. For instance, Mnih et al. [2015] use 49 games, while Van Hasselt et al.  
1115 [2016], Wang et al. [2016] use 57. Moreover, for the hyper-parameter tuning, sometimes papers use the  
1116 entire suite of games as the validation set [Bellemare et al., 2013], while in other cases hyperparameters  
1117 are optimized on a per-game basis [Jaderberg et al., 2016].

1118 Yet another inconsistency is in reporting the results in terms of the variety of different summary statistics  
1119 used to describe them, which makes direct comparisons between ideas difficult [Machado et al., 2018].  
1120 To make matters worse sometimes sufficient statistics to make a judgment on the quality of the models  
1121 are not provided. As an example, in [Bellemare et al., 2013], the main results are reported as the average  
1122 performance of the method as well as the best run without mentioning the variance or the standard error  
1123 of the mean. This is particularly problematic for reinforcement learning, where it has been shown that

<sup>16</sup>We have manually checked the papers presented ideas to improve pretrained LMs with more than 500 citations that and in all these papers GLUE has been used for evaluation.

<sup>17</sup>When predicting the action given the state, it is often done for every  $k$ -th frame, where  $k$  is the skipframe hyper-parameter.

often the variance between runs can be so large as to create statistically different distributions just by varying random seeds [Henderson et al., 2018].

The final contentious aspect of ALE that we highlight is the way that various publications choose to inject stochasticity into the environment. ALE is fully deterministic, thus it is possible to get good scores by simply memorizing the “right” action sequence, rather than learning to make good decisions in a variety of game scenarios (i.e. learning an open-loop policy). With this in mind, to encourage and evaluate agent robustness, various ideas were developed to add forms of stochasticity to ALE [Bellemare et al., 2013]. Unfortunately, these methods are not necessarily consistent with each other.

## E Popular public benchmarks for evaluating recommend systems

Table 3 presets the list of publicly available datasets for recommender systems used by the community for evaluation.

Table 3: List of popular offline datasets used for evaluating recommender systems.

Dataset	Number of examples	Users	Items	Sparsity
MovieLens 1M <sup>18</sup>	1,000,209	3706	6040	95.53%
MovieLens 20M <sup>19</sup>	13,501,622	138,159	16,954	99.42%
Amazon Product Review (Movies & TV) <sup>20</sup>	505K	22,147	178,086	99.98%
Amazon Product Review (Video Games)	46K	2,670	47,063	99.96%
Yahoo Movies <sup>21</sup>	221,367	7,642	11,915	99.76%
Pinterest <sup>22</sup>	1.5M	9916	55187	99.73%
Xing <sup>23</sup>	1,450,300	65,347	20,778	99.89%
Taobao <sup>24</sup>	100M	968K	4M	99.98%
Last.FM <sup>25</sup>	42,346	1,872	3,846	99.41%
Book-Crossing <sup>26</sup>	172,576	19,676	20,003	99.96%

1135

<sup>18</sup><https://grouplens.org/datasets/movielens/1m/>

<sup>19</sup><https://grouplens.org/datasets/movielens/20m/>

<sup>20</sup><https://jmcauley.ucsd.edu/data/amazon/>

<sup>21</sup><https://webscope.sandbox.yahoo.com/catalog.php?datatype=r>

<sup>22</sup><https://paperswithcode.com/dataset/pinterest>

<sup>23</sup><http://www.recsyschallenge.com/2017/>

<sup>24</sup><https://tianchi.aliyun.com/dataset/dataDetail?dataId=649>

<sup>25</sup><http://ocelma.net/MusicRecommendationDataset/>

<sup>26</sup><http://www2.informatik.uni-freiburg.de/~chiegler/BX/>

## 1136 **F Benchmarking checklist for the review process**

1137 This section presents a proposal for a checklist that can be used in the review process with the hope of  
1138 reducing the benchmark lottery effect. Note that as we discussed in the paper, the benchmark lottery  
1139 effect can be rooted in various aspects. As an example, Gebre et al. [2018] provided a list of questions  
1140 for the benchmark creation process that covers motivation of the benchmark, composition, collection  
1141 process, and recommended uses. Such data can be also framed as checklists for benchmark creation.

### Benchmarking checklist for reviewers and area chairs

- ☐ If there is written dissatisfaction about the author's choice of baselines, tasks, or benchmarks in the reviews, are there rationals beyond the fact that these requested datasets are "must-have" benchmarks?
- ☐ Are the reviews considering potential benefits like efficiency, fairness, and simplicity of the proposed model outside the commonly evaluated performance metrics (e.g., accuracy)?
- ☐ Are there any negative points in the reviews due to the paper proposing a method that deviates from the current trend/hype. If so, are there rational justifications for this?
- ☐ If the reviews penalizing the paper due to the proposed method not performing well only on a subset of tasks, is there enough logical elaboration on such criticism in the reviews?
- ☐ Are the reviews assessing the evaluation strategy in terms of studying the effect of different sources of variance (e.g., multiple splits, multiple random seeds, etc.)?
- ☐ If there are analyses on statistical significance testing, are they appreciated in the reviews? If there is no such analysis, are there recommendations on this provided in the reviews?
- ☐ If the paper is claiming SOTA or improvements over baselines on a benchmark, are there ablations on how much such improvement is secured by the tricks that are not tied to the main contributions?
- ☐ If the reviews are asking for more experiments, analysis, or evaluation on more benchmarks, are the potential blockers are considered for such requests? E.g. those experiments being out of reach in terms of computing budget (pre-training or extremely large datasets).
- ☐ If the paper is proposing a new idea while deviating from the common paradigms, is the "out of the hype" thinking valued in the reviews as opposed to solely recognizing SOTA performance?

1142