# SCALING LAWS FOR PRE-TRAINING AGENTS AND WORLD MODELS

Anonymous authors

004

010 011

012

013

014

015

016

017

018

019

021

023

025

051

052

Paper under double-blind review

## ABSTRACT

The performance of embodied agents has been shown to improve by increasing model parameters, dataset size, and compute. This has been demonstrated in domains from robotics to video games, when generative learning objectives on offline datasets (pre-training) are used to model an agent's behavior (imitation learning) or their environment (world modeling). This paper characterizes the role of scale in these tasks more precisely. Going beyond the simple intuition that 'bigger is better', we show that the same types of power laws found in language modeling (e.g. between loss and optimal model size), also arise in world modeling and imitation learning. However, the coefficients of these laws are heavily influenced by the tokenizer, task & architecture – this has important implications on the optimal sizing of models and data.

1 INTRODUCTION

Much progress in AI in the early 2020's has been driven by increasing model size, dataset size, and training compute. Whilst conceptually simple, the importance of this practice has led to an emerging subfield studying the *science of scaling*. This field answers questions such as how to estimate the benefit of increased compute investment, or how to optimally trade-off model and dataset size.

The role of scale in pre-training is until now best understood in the context of large language models (LLMs). Following the observation that the empirical relationship between loss and key scaling quantities can be accurately described by power laws (Kaplan et al., 2020), ensuing work studied the precise trade-off between model and dataset size (Hoffmann et al., 2022), as well as considerations about inference compute (Sardana and Frankle, 2023), repeated training data (Muennighoff et al., 2024), parameter counting (Pearce and Song, 2024), and more (Section 2).

In comparison, less is understood about scaling in *embodied AI*. Recent high-impact works suggest increasing model and dataset size can lead to ever more capable agents for two pre-training objectives; behavior cloning (BC) (Reed et al., 2022; Baker et al., 2022; Brohan et al., 2023) and world modeling (Hafner et al., 2020; Hu et al., 2023; Yang et al., 2023; Bruce et al., 2024). Such works typically demonstrate the benefit of scale through ablations over only a few model sizes, shown in terms of downstream agent performance, confirming the intuition that 'bigger is better' (Sartor and Thompson (2024) provide an aggregated analysis). However, this leaves a large gap to the precise understanding of scale in LLMs, where for a given increase in compute, models can be sized optimally, and their expected performance accurately predicted.

This paper helps close this gap. Similar to the initial study of scale in LLMs, we focus on the effect of scaling on a *generative pre-training loss* (rather than on downstream agent performance, or rewardor representation-centric objectives), in the infinite data regime, on a fixed offline dataset. Under this setting, we train families of transformers on next-token prediction tasks using architectures popular in both world modeling and BC tasks. This leads to several contributions, summarized in Figure 1.

- Scaling laws similar to those in LLMs can be observed in world modeling with tokenized observations and actions (Section 4.1, Figure 1a).
- The optimal trade-off between model and dataset size in world modeling is influenced by the tokenizer's compression rate (number of tokens per observation) (Section 4.1, Figure 1a & b).



Figure 1: This paper observes that scaling laws, as originally found in LLMs, also emerge in the tasks of world modeling and BC, when studying the pre-training loss on large datasets of human behavior. a, b) For world modeling, the power law coefficient determining optimal model size is affected by the compression rate of the tokenizer. c) In BC with tokenized image observations (BC-Token), small models need a large FLOPs budget to saturate, making these scaling laws less clear cut. d) However, moving to a single continuous embedding per observation remedies this (BC-CNN), producing prototypical scaling laws and a more balanced optimal model size coefficient.

- Scaling laws for BC with tokenized observations are harder to observe under modest compute budgets. The optimal trade-off favors smaller models and more data (Section 4.2, Figure 1c).
- Scaling laws similar to those in LLMs can once again be observed in BC with one continuous encoding per observation (Section 4.2, Figure 1d).
- Our findings can be understood through small-scale language modeling experiments (Section 5).

**Organization.** Section 2 provides detailed related work, contrasting the current understanding of scaling in embodied AI with other domains. Section 3 introduces details for our main experiments, including the architectures considered and details of scaling laws analyses. Section 4.1 & 4.2 present our main results in world modeling and BC respectively. Section 5 presents insights behind our main results, including a set of tiny-scale language experiments mimicking aspects of our main experiments. Section 6 discusses our findings and notes limitations in our work.

# <sup>108</sup> 2 RELATED WORK

110

Scaling laws origin. The term *scaling laws* is used throughout the engineering and physical sciences to denote power law relationships between two quantities, e.g. duration of a volcanic eruption and the probability of it continuing (Cannavò and Nunnari, 2016). The name derives from the *scale-invariant*<sup>1</sup> property of power laws. While early work suggested that power laws could be good empirical descriptors of important variables in deep learning (Hestness et al., 2017; Rosenfeld et al., 2019), it was Kaplan et al. (2020) who provided a comprehensive study of power laws in transformer LLMs, and popularized the usage of *scaling laws* in this context.

Scaling laws in LLMs. As the real-world value of LLMs was understood, scaling in LLMs became a high-priority research topic. Hoffmann et al. (2022) conducted a precise analysis into the trade-off of model and dataset size, finding they should be increased in equal proportions. This conflicted with Kaplan et al.'s suggestion that model size should be prioritized – an incorrect conclusion that Pearce and Song (2024) showed largely arose from counting only non-embedding parameters.

123 Many other aspects of LLM scaling analyses are beginning to be refined. Su et al. (2024) revis-124 ited the methodology used to find scaling coefficients. Hägele et al. (2024) found that multiple 125 independent cosine schedules could be reproduced more efficiently through a constant learning rate 126 with multiple short decays, or stochastic weight averaging. Pearce and Song (2024) & Porian et al. 127 (2024) found that well-tuned constant learning rates were sufficient to recover certain coefficients. 128 Bi et al. (2024) study the effect of various hyperparameters on scaling. Muennighoff et al. (2024) looked at repeated epochs, finding up to four epochs produce negligible departures from the infinite 129 data regime. Sardana and Frankle (2023) factored in inference compute to the definition of what is 130 compute-optimal. Isik et al. (2024) study the link between pre-training loss and downstream per-131 formance. A further line of research aims to explain why power laws are such a good descriptor of 132 empirical deep learning (Hutter, 2021; Maloney et al., 2022; Bahri et al., 2024). 133

Scaling in embodied AI. Compared to LLMs, our understanding of scale in embodied settings is
less advanced. Early successes in competitive games showed that running reinforcement learning
(RL) at scale could surpass human performance, e.g. (Silver et al., 2017; Berner et al., 2019). In
self-play RL, power laws were observed between certain quantities by Neumann and Gros (2022).
Meanwhile, Hilton et al. (2023) noted that, in general, reward signals do not follow power laws, and
defined a transformation of reward (intrinsic performance) that created self-consistent scaling laws.

Inspired by the effectiveness of scaling in LLMs, embodied AI research has recently begun to ex-140 plore the effectiveness of generative pre-training objectives on offline datasets, when executed at 141 scale. This includes behavior cloning objectives in video games (Baker et al., 2022; Raad et al., 142 2024), robotics (Brohan et al., 2022; 2023; Padalkar et al., 2023; Bousmalis et al., 2023), or multiple 143 domains (Reed et al., 2022), as well as world modeling objectives (Hu et al., 2023; Yang et al., 2023; 144 Bruce et al., 2024). In these studies, the benefit of scale is generally shown through increasing model 145 size on a specific downstream task of interest (e.g. measured by completion rate) – an aggregated 146 survey is provided by Sartor and Thompson (2024). 147

Whilst such studies provide strong evidence that scale can be effective in embodied domains, the 148 complexity introduced by downstream task evaluation makes quantifying the effects, and asking 149 more nuanced questions very challenging (e.g. how to trade-off model and dataset size). To ad-150 dress this, we argue that a better starting point for studying scale in these generative pre-training 151 approaches, is to study the effect of scale on pre-training loss. Whilst downstream task performance 152 may be of ultimate interest, LLM research evidences the utility of focusing on this clean intermedi-153 ate signal, which is more straightforward to analyze. Few prior works have taken such an approach. 154 In Appendix A we contrast our BC efforts with Tuyls et al. (2023) who provide a valuable initial 155 study on the relationship between compute and pre-training loss for BC. We also discuss scaling 156 analyses of image and video modeling, which are related to world modeling.

- 157
- 158
- 159 160 161

<sup>&</sup>lt;sup>1</sup>For two variables x & y, the power law  $y = ax^b$  is invariant to scaling x by a constant c. Formally:  $a(cx)^b = c^b ax^b \implies c^b y = c^b ax^b \implies y = ax^b$ .



Figure 2: The World Modelling (WM) and Behavior Cloning (BC) tasks & architecture combinations considered in this work. The fire symbol signifies trainable components, the ice symbol signifies frozen pre-trained components.

3 Methodology

This section provides details for our main experiments. We describe the pre-training tasks, architectures, and datasets considered. We also detail the methodology used in the scaling law analyses.

3.1 TASKS

171

172

173

174 175 176

177 178

179

181

182

193

194

199

200 201

202

203

204

209

We consider trajectories constructed as sequences of alternating observations  $\mathbf{o}_t$  and actions  $\mathbf{a}_t$  for timestep  $t \in \mathbb{N}$ . In this work, observations are always images,  $\mathbf{o}_t \in \mathbb{R}^{3 \times w \times h}$  and any continuous actions are discretized during preprocessing leaving,  $\mathbf{a}_t \in \{0, 1\}^{d_a}$ .

Given this data format, we consider two tasks. *World modeling* (WM) (Ha and Schmidhuber, 2018)
predicts future observations from previous observations and actions. This allows an agent to explicitly understand how its environment works, which can be used for planning, or dyna-style RL (Sutton, 2018). *Behavior cloning* (BC) predicts the future actions that the dataset's demonstrators take (Bakker et al., 1996). This creates a policy that can be directly used to act in the environment, either as-is or following further fine-tuning. Concretely, these two tasks require modeling the following quantities,

World modeling: 
$$P(\mathbf{o}_{t+1}|\mathbf{o}_t \dots \mathbf{o}_{t-k}, \mathbf{a}_t \dots \mathbf{a}_{t-k}),$$
 (1)

Behavior cloning: 
$$P(\mathbf{a}_t | \mathbf{o}_t \dots \mathbf{o}_{t-k}, \mathbf{a}_{t-1} \dots \mathbf{a}_{t-k-1}).$$
 (2)

This work focuses on *generative* pre-training aiming to model this full conditional probability distribution. We leave a study of scaling laws for alternative objectives, e.g., explicitly targeting representation learning (Nair et al., 2022) or reward-centric models (Hafner et al., 2020), to future work.

## 3.2 ARCHITECTURES

All experiments revolve around GPT-2 style causal transformers (Radford et al., 2019) as the core of the model. However we consider two different methods for inputting image observations, summarized in Figure 2. Section 3.4 details how we measure the model size of each.

**Tokenized architecture.** Our first architecture tokenizes each image observation into multiple discrete tokens. This is done with a frozen VQGAN encoder  $\text{Enc}_{\theta}(\mathbf{o}_t) \rightarrow \mathbf{z}_t$ , where  $\mathbf{z}_t \in \{1, 2, ..., V_o\}^{d_z}$ , for vocabulary size  $V_o$  and latent dimension  $d_z$ . Discretized actions are mapped to a non-overlapping vocabulary. Following tokenization, training sequences take the form,

$$[z_t^1, z_t^2, ..., z_t^{d_z}, a_t^1, a_t^2, ..., a_t^{d_a}, z_{t+1}^1, z_{t+1}^2, ..., z_{t+1}^{d_z}, a_{t+1}^1, a_{t+1}^2, ..., a_{t+1}^{d_a}, ...],$$
(3)

where each item of the sequence is an integer within our vocabulary. A transformer is then trained to maximize the likelihood of either the latent image tokens (world modeling), or action tokens (BC).

This tokenized architecture is widely used both in world modeling (Micheli et al., 2022) and BC tasks (Bousmalis et al., 2023). Gato (Reed et al., 2022) used a similar design but with continuous patches rather than discrete tokens. Our implementation tests both a 'small' (28M parameters,  $d_z = 256$ ) and 'large' (150M parameters,  $d_z = 540$ ) VQGAN – further details in Appendix B. 216 **CNN architecture.** Our second architecture differs in two ways. 1) Each image observation is 217 input into the transformer as a single continuous embedding, extracted from a small trainable con-218 volutional neural network (CNN). 2) Action dimensions are predicted independently (rather than in 219 series), assuming  $P(\mathbf{a}_t | \dots) \approx \prod_{i=1}^{d_a} P(a_t^i | \dots)$ . A single forward pass of the transformer is needed 220 per action prediction.

This produces an architecture similar to Baker et al. (2022) (VPT additionally used a transformer-XL and a refined hierarchical action space). Our implementation uses an Impala-style (Espeholt et al., 2018) CNN with 0.6M parameters for embedding image observations.

3.3 DATASETS

221

222

224 225

226 227

228

229

230

231 232

233

250

253

254

255

256

257

258

259

260 261

262

264

267

This paper focuses on the effect of scaling on the pre-training loss over an offline dataset. To study this cleanly, datasets must meet two criteria.

- 1. **Dataset size.** Repeated training on the same data alters the effect of scaling. Therefore, datasets should be large enough that all model sizes use a low number of training epochs.
- 2. **Dataset diversity.** Both the behavior and environment must contain enough richness and variety that pre-training loss does not saturate across the model sizes tested.

234 Many existing benchmark datasets fail to fulfill these criteria - if not due to limited size, then because 235 behavior is generated from a pre-trained fixed policy, or the environment is too simple. 236

237 Our work instead focuses on a dataset of human behavior collected in a video game named Bleeding *Edge*. This is a fast-paced 4 vs 4 multiplayer game, with a range of characters, abilities and maps. 238 Game play is highly complex due to the cooperative and competitive dynamics. Success requires 239 selecting high-level strategies (e.g. choosing which map regions to fight for), as well as fine-grained 240 reactive control during combat. Figure 9 shows example sequences from our dataset. 241

242 Supported by the game's developer Ninja Theory, we compiled a dataset of 8.6 years of anonymized 243 game play, containing both image observations and controller actions. We refer to this as the 7 map 244 *dataset*. We also use a subset of this for some experiments, of around 1.1 years from a single map, 245 which we name the Sky Garden dataset. Appendix B.3 provides further technical details.

246 247 3.4 SCALING ANALYSIS METHODOLOGY

248 We are interested in studying the relationship between several quantities defined below. 249

- Model size N, the total number of trainable parameters (ignoring VQGAN parameters for WM-Token & BC-Token, but including the fixed-size CNN for BC-CNN). Embedding parameters are included in the count following Pearce and Song (2024).
- Dataset size D, the total number of *inputs* the transformer sees during training. For WM-Token and BC-Token this is  $d_z + d_a$  per observation & action pair, and for BC-CNN this is one per observation & action pair.
- Compute C, the number of floating point operations (FLOPs) used during training. The common approximation of C = 6ND (Kaplan et al., 2020) is used.
- Loss L, the standard classification cross-entropy loss (all targets are discretized). We assume training loss is an accurate proxy for test loss (Appendix B.3.1 analyzes further).

More specifically, we are interested in 'compute-optimal' versions of each quantity. For loss, this is defined as the minimal loss possible for a given FLOPs budget,

$$L_{\text{optimal}}(C) = \min_{\text{s.t. } C = 6ND} L(N, D), \tag{4}$$

265 where L(N, D) is the empirical loss achieved with an N parameter model trained on D tokens. We 266 further define optimal model and dataset sizes as the configuration that produce this minimal loss given a FLOPs budget, 268

$$N_{\text{optimal}}(C), D_{\text{optimal}}(C) = \underset{N,D \text{ s.t. } C=6ND}{\operatorname{argmin}} L(N, D).$$
(5)

Table 1: Summary of fitted scaling coefficients for our main experiments. Note that we favor the Frontier fit when available, and only use the Parametric fit for BC-Token-540 (see Section 3.4).

	Front	ier fit	Parametric fit			
Experiment	$N_{\rm optimal} \propto C^a$	$D_{\rm optimal} \propto C^b$	$N_{\rm optimal} \propto C^a$	$D_{\rm optimal} \propto C^b$		
WM-Token-256	0.49	0.51	0.52	0.48		
WM-Token-540	0.62	0.37	0.78	0.22		
BC-Token-540	N/A	N/A	0.32	0.68		
BC-CNN	0.66	0.34	0.47	0.53		

283

284 285 286

298 299 300

305 306 307

270

**Scaling analysis.** The heart of scaling law analysis is fitting power law relationships predicting these compute-optimal quantities. For predicting optimal model and dataset size, we use,

$$\hat{V}_{\text{optimal}}(C) = a_0 C^a \qquad \hat{D}_{\text{optimal}}(C) = b_0 C^b, \tag{6}$$

with fitted constants  $a_0, a, b_0, b$ . <sup>2</sup> We consider two methods to fit these relationships, introduced by Hoffmann et al. (2022). Their Method 1, which we term *Frontier fit*, classifies efficient models as those falling on the *efficient frontier* (see Figure 1). Coefficients can then be estimated straightforwardly through a line of best fit on a plot of FLOPs vs parameters or data for these efficient models.

Frontier fit is our preferred method when available – it avoids making any assumptions about the
training curves, directly fitting the best models observed. However, it requires training models past
the point where they are the optimal configuration (seen on a loss-FLOPs plot as overlapping curves).
In some of our experiments (BC-Token and Section 5.1), this was not possible.

In these situations, we resort to Method 3 of Hoffmann et al. (2022), which we term *Parametric fit*. This fits the coefficients  $\alpha$ ,  $\beta$ ,  $N_c$ ,  $D_c$ , E to a parametric loss form,

$$\hat{L}(N,D) = \frac{N_c}{N^{\alpha}} + \frac{D_c}{D^{\beta}} + E,$$
(7)

to the empirical training curves. In our implementation, we use SciPy's curve\_fit function. We then find  $a = \beta/(\alpha + \beta)$ ,  $b = \alpha/(\alpha + \beta)$ . This makes a very strong assumption about the training curves, but allows coefficients to be estimated at a smaller compute budget.

For loss prediction we use the form recommended by Pearce and Song (2024),

$$\hat{L}_{\text{optimal}}(N,D) = c_0 C^{-c} + E.$$
(8)

We again use the curve\_fit function, fitted to models along the efficient frontier. During fitting, we set bounds on the variables:  $c_0 \in [0, \infty], c \in [-1, 1], E \in [0.1, \infty]$ .

Training details. While early scaling studies conducted sweeps over multiple cosine decays of differing lengths (Kaplan et al., 2020; Hoffmann et al., 2022), follow up work found this redundant (Pearce and Song, 2024; Hägele et al., 2024; Porian et al., 2024). We follow the approach of using a constant learning rate per model, so each requires only one training run. We aim to train models until they have passed their compute efficient FLOPs budget. We only modify the parameters of the transformer, following the configurations documented in Appendix B.

316 317 318

319

## 4 SCALING ANALYSIS IN EMBODIED AI

This section presents our main results. We begin by considering the scaling laws for the task of
 world modelling in Section 4.1 with two different tokenizers (turning image observations into 256
 and 540 tokens for the *small* and *large* variants respectively). Section 4.2 then considers the task of
 BC both with *tokenized* and *CNN* architectures. Finally, Section 4.3 tests the extrapolation capability of these scaling laws for the task of world modeling.



Figure 3: WM-Token scaling with  $d_z = 256$  tokens per image observation. Left shows the parametric fit. Middle & right show the *frontier fit* estimating optimal model & dataset size respectively.



Figure 4: WM-Token scaling with  $d_z = 540$  tokens per image observation. Left shows the parametric fit. Middle & right show the frontier fit estimating optimal model & dataset size respectively. Compared to the results for WM-Token-256, the power law coefficient for  $N_{\text{optimal}}$  increases from 0.49 to 0.62.

#### 4.1SCALING ANALYSIS IN WORLD MODELING

Figures 3 & 4 present our results for the task of world modeling, with the scaling law coefficients summarised in Table 1. For WM-Token-256 we find that the optimal coefficients for model and dataset size are both  $\approx 0.5$ , e.g. one should increase both model and dataset size in the same proportions. This matches the scaling laws observed in LLMs (Hoffmann et al., 2022). Increasing the number of tokens per image to 540 for WM-Token-540 changes the optimal trade-off between model and dataset size, skewing towards model size;  $N_{\text{optimal}} = 0.62$ ,  $D_{\text{optimal}} = 0.37$ . We discuss this further in Section 5.3. 

4.2 SCALING ANALYSIS IN BEHAVIOR CLONING



Figure 5: BC-Token scaling with  $d_z = 540$  tokens per image observation. Models above 2M param-eters do not saturate over the FLOPs range considered and coefficients can not be inferred using the frontier fit method. 

We present our results on the scaling law coefficients for BC-Token in Figure 5. Despite sharing an architecture with WM-Token-540 we now observe the opposite dependence on model and dataset

<sup>2</sup>Note that by subscribing to C = 6ND we find a = 1 - b;  $N \propto C^a \implies C/D \propto C^a \implies D \propto C^{1-a}$ .



Figure 6: BC-CNN scaling. Left shows the *parametric fit*. Middle & right show the *frontier fit* estimating optimal model & dataset size respectively. Compared to the results for BC-Token the model sizes considered compute-optimal are considerably larger. The power law coefficient for  $N_{\text{optimal}}$  also increases significantly from 0.32 to 0.66 skewing towards scaling model size as opposed to dataset size when scaling up compute.

391 392

406

407

408

410 411 412

413

414

415 416 417

418

419

387

388

389

390

393 sizes. The coefficients skew heavily towards dataset size;  $N_{\text{optimal}} = 0.32$ ,  $D_{\text{optimal}} = 0.68$  (com-394 pared to  $N_{\text{optimal}} = 0.62$ ,  $D_{\text{optimal}} = 0.37$  – explained in Section 5.1). Furthermore, under the same 395 compute budget the compute-optimal model sizes are significantly smaller. For a compute budget of 396 10<sup>18</sup> and 10<sup>19</sup> FLOPs we find that model sizes of 2M and 11M are compute-optimal for BC-Token-397 540 compared to 27M and 110M for WM-Token-540. In our experiments, we observe the losses 398 for the BC-Token models take much longer to plateau leading to less overlap between model sizes. 399 This results in the *frontier fit* not being suitable for accurately estimating the scaling law coefficients, hence we rely on the *parametric fit* for these results. 400

To better understand the change in the scaling law coefficients, we now consider the BC-CNN architecture for the task of BC in Figure 6. For this architecture, we observe that the coefficients now skew towards model size (similarly to those in (Tuyls et al., 2023)), with  $N_{\text{optimal}} = 0.66$ , and  $D_{\text{optimal}} = 0.34$ . Section 5.2 provides more intuition on the differences between the WM-Token and BC-Token setups that lead to this change.

Further to studying the differences in scaling law coefficients between tasks and architectures, we also study the accuracy of extrapolation.

### 4.3 EXTRAPOLATION IN WORLD MODELING



Figure 7: Testing the extrapolation capability of our derived scaling law for WM-Token-256 by
training an 894M parameter model with an order of magnitude more compute than was used for
the scaling law analyses. We observe good agreement between our predicted optimal loss/model
size/number of training tokens (dotted lines) and our actual training run.

424

To test the extrapolation accuracy of our derived scaling laws, we train a 894*M* parameter WM-Token-256 model with an order of magnitude more compute than used for the scaling law analyses. Figure 7 presents both the learning curve as well as the extrapolated lines derived from the Frontier fit method. We take the point with the loss value closest to our extrapolated loss curve ( $\sim 1.58 \times 10^{21}$ FLOPS), and mark it on the Frontier fit extrapolations. We observe very good agreement between that point and our compute-optimal predictions for both model and dataset size, demonstrating the accuracy of our derived scaling laws. The gap between our prediction and the actual training run suggests we could further optimize the hyperparameters (learning rate and batch size in particular) for the 894M parameter model, which was not extensively tuned due to compute requirements.

5 FURTHER ANALYSIS

435

436 437

438

439

440 441

442

443

444

445

446

Section 4 made several observations about the effect of scale in the pre-training of embodied agents. This section aims to understand these results further, and provide intuition for why they occur. Specifically we target three questions.

- Q1: Why does BC-Token produce training curves that do not plateau, while WM-Token does, given an identical architecture and dataset? (Section 5.1)
- Q2: Why does moving from BC-Token to BC-CNN resolve this issue? (Section 5.2)
- Q3: Why does increasing the amount of tokens per image observation (256 to 540) lead to an increase in the optimal model size coefficient (0.49 to 0.62)? (Section 5.3)
- 447 5.1 Q1: BC-Token vs. WM-Token

The lack of saturation of BC-Token models compared to WM-Token models can be attributed to two factors. The first is a sparser loss. A single observation-action pair is discretized into  $d_z + d_a$  total tokens. With the large VQGAN tokenizer, world modeling receives supervision for  $d_z/(d_z + d_a) =$  $540/556 \approx 97\%$  tokens, while BC is supervised for  $d_a/(d_z + d_a) = 16/556 \approx 3\%$  of tokens.

- The second factor is the granularity of the targets. The large tokenizer creates a world modeling vocabulary size of 4096. Each vocabulary item roughly corresponds to a specific color and texture for an image patch. Many vocabulary items may only be used to model specific map regions or special abilities. Hence, the world modeling loss is very granular. On the other hand, a player can take the same action in multiple different situations – continue straight could be used to escape an enemy, chase an enemy, or navigate to a checkpoint. Hence, the supervision for BC is more vague and abstracted. We can think of this as a *super-classed* label.
- To demonstrate the effect of these two factors on optimal model size coefficients, we run a set of tiny-scale experiments in language modeling. Transformers are trained on next-character prediction, on a dataset of Shakespeare text<sup>3</sup> using a single character for each token. Model sizes are varied from 4k parameters to 17M parameters. Context length is fixed at 16 characters/tokens.
- Figure 8 (left) shows training curves over all 16 tokens, followed by a *sparse* loss where supervision is only provided from the final token (middle), and then additionally under a *super-classed* setting (right). This super-classes the final target – rather than using all 128 ASCII characters, they are randomly shuffled into one of two macro classes.
- These modifications are intended to mirror the effect of moving from WM-Token to BC-Token. We compute optimal model size coefficients using the parametric fit as most models are not trained long enough for the frontier fit method. Indeed, we see that the coefficient drops from 0.63 to 0.15 with both the sparse and super-classed loss. This matches the magnitude of decrease seen in Table 1 from 0.66 to 0.32, indicating that the proposed mechanisms explain our findings.
- 474

485

5.2 Q2: BC-TOKEN VS. BC-CNN

Despite the same non-granular loss signal, why does switching architecture from BC-Token to BC CNN makes the loss of similar model sizes plateau under a much smaller compute budget?

Consider each architecture using a transformer with 1M parameters. Observe from Figure 2 that BC-Token receives  $d_z + d_a = 556$  inputs for every action  $\hat{a}_t$  it predicts, while BC-CNN receives just one input for every action predicted. Hence, BC-Token uses around 556 times more compute in its action prediction  $(556 \times 2 \times 1M \approx 1 \times 10^9 \text{ FLOPs})$  than BC-CNN  $(1 \times 2 \times 1M \approx 2 \times 10^6 \text{ FLOPs})$ . This means that even with the same number of parameters, BC-Token can learn a far more expressive function than BC-CNN. Hence, BC-Token requires far more tokens to match this expressivity, and training curves for a given model size plateau much later.

<sup>&</sup>lt;sup>3</sup>Shakespeare character dataset from: https://github.com/karpathy/nanoGPT



Figure 8: Training curves and parametric fit for character modeling experiments. The standard dense LLM loss has been modified to reflect properties of BC – a sparse loss (one of 16 tokens), and then additionally super-classing the targets into two classes.

5.3 Q3: WM-TOKEN-256 vs. WM-TOKEN-540

Finally, we seek to understand why the optimal model size coefficient increases when moving from the 256 to the 540 token VQGAN. As the number of tokens per image observation are increased, the compression rate of the tokenized representation decreases. We would expect that each individual token becomes easier to predict in this less compressed representation. This would mean a less expressive function is needed (smaller model size), but also a smaller number of examples would need to be seen (smaller dataset size). It is less clear what ratios these ingredients decrease in, and hence what effect a lower compression rate has on the optimal model size coefficient.

For further insight into how compression affects the optimal model size coefficient, we ran a small
scale experiment in language modeling using two text representations; 1) ASCII character-level
tokenization. (low compression) 2) GPT-2 tokenizer (high compression). We used the BookCorpus
dataset (Zhu et al., 2015), and trained models past their compute-optimal point, so the *Frontier fit*method could be used for coefficient estimation.

515 Appendix C shows results. Under the character-level tokenizer (low compression), we find 516  $N_{\text{optimal}} \propto C^{0.66}$ . For the GPT-2 tokenizer (high compression), we find  $N_{\text{optimal}} \propto C^{0.44}$ . The 517 key takeaway is that the more compressed representation has led to a lower optimal model size 518 coefficient, which reflects what we see in the world modeling experiments.

519 520

521 522

497

498

499 500 501

502

## 6 DISCUSSION & CONCLUSION

This paper establishes a deeper understanding of scaling laws for world modeling and behavior cloning, two tasks that underpin embodied AI applications in domains such as video games and robotics. Focusing on generative pre-training of such models, we show that it is possible to recover scaling laws similar to those established in the LLM literature. Establishing such a link is key to making efficient use of available resources, and to training compute-optimal models.

Considering the task of world modeling, we find that models can be smoothly scaled following
 best practices and insights from the LLM literature. Surprisingly, the scaling coefficients for our
 WM-Token-256 architecture very closely match those established for LLMs. Comparing to our
 WM-Token-540 model and additional analysis, we further establish that scaling is affected by the
 tokenizer's compression rate.

Turning to pre-training BC policies for agents, the choice of architecture is extremely important in
determining optimal scaling behavior. When using architectures with tokenized image observations,
dataset size should be increased much more rapidly than model size. Meanwhile, for BC-CNN
architectures, model size should be increased faster than dataset size.

Limitations. While we show that scaling laws can be precisely described in the infinite data regime
 and for appropriate architectures, future work is needed to establish scaling laws for alternative mod els and under varying dataset quality. In addition, we focus on loss as an intermediate quantity that
 can be effectively optimized in pre-training. Many additional considerations are required for effec-

540 tive AI models, such as downstream task performance and model inference times. How valuable 541 scaling laws can be in providing insights relevant to those choices remains an open question. 542

#### 543 ETHICS STATEMENT 544

Data for this project was provided via a partnership with Ninja Theory, who collected a large corpus 545 of human gameplay data for their game Bleeding Edge. Data collection was covered by an End User 546 License Agreement (EULA) and our use of the data was governed by a data sharing agreement with 547 the game studio, and approved by our institution's IRB. This data was recorded between September 548 2020 and October 2022. To minimize risk to human subjects, any personally identifiable information 549 (Xbox user ID) was removed from the data. The resulting data was cleaned to remove errors and 550 data from bad actors.

551 552

561

567

568

569

570

574

575

576

577

581

583

#### REFERENCES 553

- 554 Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural 555 scaling laws. Proceedings of the National Academy of Sciences, 121(27):e2311878121, 2024. 556
- Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching 558 unlabeled online videos. Advances in Neural Information Processing Systems, 35:24639–24654, 559 2022. 560
- Paul Bakker, Yasuo Kuniyoshi, et al. Robot see, robot do: An overview of robot imitation. In 562 AISB96 Workshop on Learning in Robots and Animals, volume 5, pages 3–11, 1996. 563
- Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy 564 Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large 565 scale deep reinforcement learning. arXiv preprint arXiv:1912.06680, 2019. 566
  - Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. arXiv preprint arXiv:2401.02954, 2024.
- Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauza, 571 Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving 572 foundation agent for robotic manipulation. arXiv preprint arXiv:2306.11706, 2023. 573
  - Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817, 2022.
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choroman-578 ski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action 579 models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818, 2023. 580
- Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, 582 Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In Forty-first International Conference on Machine Learning, 2024. 584
- Flavio Cannavò and Giuseppe Nunnari. On a possible unified scaling law for volcanic eruption 585 durations. Scientific reports, 6(1):22289, 2016. 586
- Lasse Espeholt, Hubert Sover, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam 588 Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with im-589 portance weighted actor-learner architectures. In International conference on machine learning, 590 pages 1407-1416. PMLR, 2018. 591
- Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image 592 synthesis. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12873-12883, 2021.

594 William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL https: 595 //github.com/Lightning-AI/lightning. 596 David Ha and Jürgen Schmidhuber. World models. arXiv preprint arXiv:1803.10122, 2018. 597 598 Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. arXiv preprint arXiv:2010.02193, 2020. 600 601 Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin 602 Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. arXiv preprint 603 arXiv:2405.18392, 2024. 604 Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo 605 Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative 606 modeling. arXiv preprint arXiv:2010.14701, 2020. 607 608 Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, 609 Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, 610 empirically. arXiv preprint arXiv:1712.00409, 2017. 611 Jacob Hilton, Jie Tang, and John Schulman. Scaling laws for single-agent reinforcement learning. 612 arXiv preprint arXiv:2301.13442, 2023. 613 614 Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza 615 Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Train-616 ing compute-optimal large language models. arXiv preprint arXiv:2203.15556, 2022. 617 Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shot-618 ton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. arXiv 619 preprint arXiv:2309.17080, 2023. 620 621 Marcus Hutter. Learning curve theory. arXiv preprint arXiv:2102.04074, 2021. 622 623 Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and 624 Sanmi Koyejo. Scaling laws for downstream task performance of large language models. arXiv preprint arXiv:2402.04177, 2024. 625 626 Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, 627 Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language 628 models. arXiv preprint arXiv:2001.08361, 2020. 629 630 Andrej Karpathy. NanoGPT. https://github.com/karpathy/nanoGPT, 2022. 631 Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining 632 Xie. A ConvNet for the 2020s, March 2022. URL http://arxiv.org/abs/2201.03545. 633 arXiv:2201.03545 [cs]. 634 635 Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. 636 arXiv preprint arXiv:2210.16859, 2022. 637 638 Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world mod-639 els. arXiv preprint arXiv:2209.00588, 2022. 640 Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra 641 Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language 642 models. Advances in Neural Information Processing Systems, 36, 2024. 643 644 Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A univer-645 sal visual representation for robot manipulation. arXiv preprint arXiv:2203.12601, 2022. 646 Oren Neumann and Claudius Gros. Scaling laws for a multi-agent reinforcement learning model. 647 arXiv preprint arXiv:2210.00849, 2022.

648 649 650	Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. <i>arXiv preprint arXiv:2310.08864</i> , 2023.
651 652 653	Tim Pearce and Jinyeop Song. Reconciling kaplan and chinchilla scaling laws. <i>arXiv preprint arXiv:2406.12907</i> , 2024.
654 655 656	Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving discrepancies in compute-optimal scaling of language models. <i>arXiv preprint arXiv:2406.19146</i> , 2024.
657 658 659 660	Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, et al. Scaling instructable agents across many simulated worlds. <i>arXiv preprint arXiv:2404.10179</i> , 2024.
661 662	Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. <i>OpenAI blog</i> , 2019.
663 664 665 666	Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. <i>arXiv preprint arXiv:2205.06175</i> , 2022.
667 668	Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. <i>arXiv preprint arXiv:1909.12673</i> , 2019.
669 670 671	Nikhil Sardana and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. <i>arXiv preprint arXiv:2401.00448</i> , 2023.
672 673	Sebastian Sartor and Neil Thompson. Neural scaling laws for embodied ai. <i>arXiv preprint arXiv:2405.14005</i> , 2024.
674 675 676 677	David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. <i>nature</i> , 550(7676):354–359, 2017.
678 679	Hui Su, Zhi Tian, Xiaoyu Shen, and Xunliang Cai. Unraveling the mystery of scaling laws: Part i. <i>arXiv preprint arXiv:2403.06563</i> , 2024.
680 681	Richard S Sutton. Reinforcement learning: An introduction. A Bradford Book, 2018.
682 683	Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. <i>arXiv preprint arXiv:2404.02905</i> , 2024.
684 685 686	Jens Tuyls, Dhruv Madeka, Kari Torkkola, Dean Foster, Karthik R Narasimhan, and Sham M Kakade. Scaling laws for imitation learning in single-agent games. <i>arXiv preprint</i> , 2023.
687 688	Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. <i>arXiv preprint arXiv:2310.06114</i> , 2023.
689 690 691 692	Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. <i>ICLR</i> , 2022.
693 694 695 696	Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In <i>The IEEE International Conference on Computer Vision (ICCV)</i> , December 2015.
697	
698 699	
700	

702 The appendix is organized as follows. 703 704 • Appendix A contains an extended related work section. 705 • Appendix B contains details on the training of the model configurations, hyperparameters, 706 and a description of the datasets used. • Appendix C contains results from Section 5.3. 708 • Appendix D contains further results training world models on a robotics task. 709 710 • Appendix E contains further results demonstrating the link between pre-training loss and 711 performance. 712 713 EXTENDED RELATED WORK А 714 715 Here we contrast more granular details of several related works. 716 717 Tuyls et al. (2023) provide (amongst other things) a valuable initial study on the relationship between compute and pre-training loss for BC. They focus on Atari and Nethack games, where the dataset is 718 generated by a fixed pre-trained agent. They test one architecture - a single-layer LSTM network, 719 scaling width-wise to increase parameter count. They find the model size coefficient  $N_{
m optimal} \propto C^a$ 720 varies between 0.58 to 0.79 for different games. 721 722 By contrast, our work focuses on transformer-based models, and we consider two variants of archi-723 tecture. We also train on human behavior, rather than pre-trained agents (which could be viewed as model distillation). Additionally, we go beyond BC and also study scale in world modeling. 724 725 Scaling laws in other domains. Scaling laws have also been observed in auto-regressive model-726 ing of other modalities. Most relevant to our world modeling task are those of video and images. 727 Henighan et al. (2020) found the optimal trade off between model and dataset size to match their reported LLM coefficient ( $N_{\text{optimal}} \propto C^{0.7}$ ) and was not affected by tokenizer. Our experiments offer 728 729 different findings in the domain of world modeling on human data - we use updated methodologies 730 to measure this trade-off (Hoffmann et al., 2022), finding it is affected by the tokenizer. Tian et al. 731 (2024) show scaling laws emerge in image modeling with non raster-ordered next-token prediction. 732 733 В SCALING EXPERIMENTS FURTHER DETAILS 734 735 **B.1** HYPERPARAMETERS 736 737 We trained two VQGANs from scratch with reconstruction losses. 738 • **BE-Small.** Based on Esser et al. (2021), uses  $d_z = 256$ ,  $V_0 = 4096$ , h = w = 128, with 739 28M parameters, and a CNN design. It was trained on a single SkyGarden Bleeding Edge 740 map. 741 • **BE-Large.** Based on Yu et al. (2022), uses  $d_z = 540$ ,  $V_0 = 4096$ , h = 180, w = 300, with 742 150M parameters, and a vision transformer design. It was trained on all seven Bleeding 743 Edge maps. 744 745 We selected the numbers of tokens per image based on qualitative assessment of reconstructions. We 746 found that 256 tokens per image was the minimum that still allowed a reconstruction to capture the 747 majority of salient gameplay details. However certain details still were lacking, such as an enemy 748 player's health bars – hence we also considered a 540 token version that provided a higher quality 749 reconstruction. 750 BC-CNN details. We use h = w = 128. The 0.6M paramter CNN is similar to that used by (Baker 751 et al., 2022), however it uses ConvNext blocks (Liu et al., 2022). The CNN produces an embedding 752 of size 1024 which is then put through a linear layer to obtain a vector matching the transformer's 753

- embedding dimension.
- 755 Transformer configurations are given in Table 2. We describe the parameters for the WM-Token architecture. Note that MLP layers are four times the width of embed dim. Model configurations

roughly followed the model configurations used in Table A9 of Hoffmann et al. (2022), where residual stream dimension, number of layers, and number of heads were roughly increased proportionally.

Table 2: Transformer configurations. Here N is listed for the tokenized architectures. Parameter count varies slightly for BC-CNN due to inclusion of the embedding CNN and differing numbers of embedding parameters sizes.

763		01	M	Lovor	Num booda	Embod dim
764				Layers	Ivuili neaus	Emped um
765			2M	3	3	180
766			4M	4	4	240
767			11M	6	6	360
768			15M	4	4	512
769			27M	8	8	512
770			52M	10	10	640
771			110M	15	12	768
772			206M	16	16	1024
773			894M	23	14	1792
774						
775						
776						
777	B.2	TRAINING DET	AILS			

All transformers are trained with a variant of nanoGPT (Karpathy, 2022) using PyTorch Lightning (Falcon and The PyTorch Lightning team, 2019).

781 This section lists key hyperparameters. Note that it was important to find optimization settings that 782 produced the lowest possible loss for a given model size. In general larger models require smaller 783 learning rates. Our approach first optimized the smallest model through a grid sweep, we would 784 then sequentially run a sweep over the next largest model, starting at the smaller model's optimized 785 learning rate. Table 3-6 provide final settings.

Table 3: Hyperparameters for WM-Token with  $d_z = 256$  tokens per image observation.

N Seq len Co		Context length	Tokens per update	Learning rate	
15M	I 10	2,720	522,240	0.0007	
27M	I 10	2,720	522,240	0.0007	
52M	I 10	2,720	522,240	0.0007	
1101	M 10	2,720	522,240	0.0007	
2061	M 10	2,720	522,240	0.00057	
8941	M 10	2,720	2M	0.00028	

Table 4: Hyperparameters for WM-Token with  $d_z = 540$  tokens per image observation.

N Seq len		Context length	Tokens per update	Learning rate	
43.6	10	5 5 60	522 560	0.005	
4M	10	5,560	533,760	0.005	
11M	10	5,560	533,760	0.001	
27M	10	5,560	533,760	0.001	
52M	10	5,560	533,760	0.001	
110M	10	5,560	533,760	0.0005	
206M	10	5,560	533,760	0.0005	

811	Table 5: Hyperparameters for BC-Token with $d_z = 540$ tokens per image observation.								
812		N	Seg len	<b>Context length</b>	Tokens per	update	Learnin	ng rate	e
813	_			8	•	•		8	_
814		2M	10	5.560		533,760		0.0005	5
815		4M	10	5,560		533,760	0.0005		5
816		11M	10	5,560		533,760	0.0001		Ĺ
817		27M	10	5,560		533,760		0.0001	L
818									
819									
820				Table 6. Hyperpa	rameters for B	C-CNN			
821		3.7	C I		T4	1 4	т ·		
822		IN	Seq len	Context length	Items per	update	Learning	g rate	_
823		214	10	10		2560	0	0002	
824		21 <b>VI</b> 2 <b>M</b>	10	10		2560	0	0003	
825		51VI 10M	10	10		2560	0	0.0003	
826		26M	10	10		2560	0	00003	
827		51M	10	10	)	2560	0	0003	
828		51101	10	10		2500	0		
829									
830			the same data	NO. CONTRACTOR OF A					Mite Harrister
831	Jersey Sink		- 371 0			2		0	5000
832								- Sec	
833	Hydrocore					1000 (CO		Cont.	
834	Aqueducts								
835						A	2	Ç.N	A
836	Sky Garden					2010 50			
837								11 11 11 11 11 11 11 11 11 11 11 11 11	Not a star of the
838	Landslide						D Z		
839						THE REAL			
840	Boneyard		11 - an		4 0				. 10
841	Facility 665								
842				100 100 100 100 100 100 100 100 100 100		2	goo (		11. Inc.
843		<i>t</i> =0	<i>t</i> =1	t=2 t=3	<i>t</i> =4 <i>t</i> =5	<i>t</i> =6	<b>t</b> =7	<b>t</b> =8	<i>t</i> =9
977					Timesteps				

Figure 9: Example trajectories from a dataset of 8.6 years of human gameplay in the video game Bleeding Edge across 7 maps.

847 848 849

850

851

845

846

810

# **B.3** DATASET DETAILS

852 Image observations were stored in MP4 format at 60fps, alongside binary files containing the asso-853 ciated controller actions. A time code extracted from the game was stored for each frame, to ensure actions and frames remained in sync at training time. 854

855 The 7 Maps dataset comprised 60,986 matches, yielding 530,713 individual player trajectories (each 856 around 9 minutes), totaling 27.89 TiB on disk. This amounted to around 8.6 years of gameplay. After 857 downsampling to 10Hz (the frequency models are trained on), this equated to 1.63B frames. This 858 was then divided into training / validation / test sets by dividing the matches with an 80:10:10 split.

859 Our filtered Sky Garden dataset used the same 80:10:10 split and 10Hz downsampling, but focused 860 on just one map, yielding 71,940 individual player trajectories, or 355.5M frames (around 1.12 years 861 of game play). 862

For discretizing the controller actions, while the buttons are natively discrete, we discretize the x 863 and y values of the left and right joysticks into eleven buckets.

# 864 B.3.1 ON THE INFINITE DATA REGIME

We wish to study scaling in the infinite data regime, where training loss is not significantly effected
by models repeatedly training on the same datapoints which can lead to overfitting effects. This
section calculates the number of training tokens allowed for each model family trained in this work.
Viewing Figure 1 alongside these numbers confirms that models remain in the infinite data regime
for all our experiments.

**WM-Token-540, BC-Token-540.** We trained on the 7 maps dataset, with 1.63B observation-action pairs. Models used the tokenized architecture with the large VQGAN, so each observation-action pair creates 540 + 16 = 556 transformer inputs, for a total of  $1.63B \times 556 = 906B$  training tokens. Muennighoff et al. (2024) observe that tokens may be reused up to four times with negligible departure from the infinite data regime. This produces 3.6T tokens. For a 200M parameter model the compute allowed by the infinite data regime is  $C = 6ND = 6 \times 200M \times 3.6T = 4.3 \times 10^{21}$  FLOPs.

877 WM-Token-256. This is trained on the *Sky Garden* dataset, with 355M observation-action pairs. 878 Each pair is split into 256 + 16 = 272 tokens, for 97B training tokens, or  $97B \times 4 = 386B$  effective 879 tokens. For a 200M parameter model the compute allowed by the 'infinite data regime' is  $C = 6ND = 6 \times 200M \times 386B = 4.6 \times 10^{20}$  FLOPs.

**BC-CNN.** Trained on 7 maps dataset, but now with one token per observation-action pair, this creates a possible  $1.63B \times 4 = 6.52B$  effective tokens. A 50M parameter model uses  $C = 6ND = 6 \times 50M \times 6.52B = 2.0 \times 10^{18}$  FLOPs.

# C FURTHER ANALYSIS DETAILS

Experimental results supporting Section 5.3.



Figure 10: Relating to Section 5.3, character-level (low compression). Utilising the *frontier* fit (middle and right) we derive the power law coefficient for  $N_{\text{optimal}}$  as 0.66 and  $D_{\text{optimal}}$  as 0.34.



Figure 11: Relating to Section 5.3, GPT-2 tokenizer (high compression). Utilising the *frontier* fit (middle and right) we derive the power law coefficient for  $N_{\text{optimal}}$  as 0.44 and  $D_{\text{optimal}}$  as 0.56, an increase from 0.66 in Figure 10 found when utilising a lower compression character-level tokenizer.

#### D WORLD MODELING FOR ROBOTICS

To supplement our main experiments conducted on a large dataset of human behavior in a video game, we conduct smaller scale experiments on the real-world robotics RT-1 dataset (Brohan et al., ) – a dataset of high-skill human demonstrations. These firstly evidence that scaling laws emerge in other environments. Furthermore, we find further support that tokenizer compression rate (number of tokens per observation) affects scaling coefficients.

We use the RT-1 dataset, resized to 128x128 pixels per image. We trained a set of five VQVAEs with  $z_o \in [16, 36, 64, 100, 256]$  and  $V_o = 4096$  for 40,000 updates on batches of 128. Reconstructions are visualized in Figure 12.



Figure 12: VQVAE reconstructions on the RT-1 dataset for differing numbers of tokens per observation,  $z_o \in [16, 36, 64, 100, 256]$ .

For each VQVAE, we then train a range of WM-Token model sizes N $\in$ [0.08M, 0.2M, 0.28M, 0.54M, 0.99M], and measure scaling coefficients using the frontier fit method - see Figure 14 for training curves.

First, we observe that scaling laws similar to those in our main set of experiments emerge for all VQ-VAEs, with predictably decreasing losses and similar optimal parameter scaling coefficients (0.56 to 0.65 in Figure 14). 

Secondly, we observe that the optimal parameter scaling coefficient increases with increasing num-bers of tokens per image. To verify this, we repeated three times, and in Figure 13 plot all coeffi-cients. This supports our claim that a higher compression rate leads to a lower optimal parameter coefficient. 



Figure 13: RT-1 experiments. Optimal parameter coefficient vs. number of tokens per observation, with three repeated runs per VQVAE.



Figure 14: RT-1 experiments. Note that the optimal parameter coefficient increases with the number of tokens per observation.

# 1026 E PRE-TRAINING LOSS AND PERFORMANCE

This section presents evidence for pre-training loss correlating with agent performance.

# 1030 E.1 PRE-TRAINING LOSS VS. WORLD MODELING METRICS

We use metrics commonly used to assess the quality of the world models (Yang et al., 2023), originally developed in the video generation literature. Conditioned on an initial real frame and a sequence of real actions, we compare the observations generated by the world model, with the real sequence of observations, measuring FVD, LPIPS and PSNR. Specifically, we generate 1024 videos each of 10 seconds. We perform this for various checkpoints on each size in our WM-Token-256 set of models. This allows a plot of the checkpoint pre-training loss vs video generation metric to be assessed.

Figure 15 shows results. We find correlations of 0.77, -0.57, 0.83 for LPIPS, PSNR, FVD respectively. This evidences the strong relationship between pre-training loss and world model quality.



1073Figure 15: WM-Token-256 pre-training loss vs. video generation metrics. Pearson correlation of10740.77, -0.57, 0.83 is found for LPIPS, PSNR, FVD respectively.

1077 E.2 BEHAVIORAL CLONING META-ANALYSIS

1079 We conducted a meta-analysis of Tuyls et al. (2023) to quantify the strength of relationship between pre-training loss and online return. This is possible in their set up for two reasons. 1) Tuyls et al.

use a simple environment that can be run cheaply as a gym environment, so all checkpoints can be
 evaluated. 2) Tuyls et al.'s data generating policy is an expert rules-based policy, so BC necessarily
 leads to an expert agent.

By pairing data points from Figure 5a and b of Tuyls et al. (2023), we plot pre-training loss vs. online return in Figure 16. This produces a correlation coefficient of -0.97, which provides strong evidence that in the infinite data regime, pre-training loss is a good proxy for online performance.



Figure 16: Meta analysis of Tuyls et al. (2023), assessing strength of the relationship between pre training loss and online return. This came out to a Pearson correlation coefficient of -0.97.



1130 1131

1132