Scaling Laws for Pre-training Agents and World Models

Tim Pearce [*1] **Tabish Rashid** [*1] **David Bignell** [1] **Raluca Georgescu** [1] **Sam Devlin** [1] **Katja Hofmann** [1]

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 also arise in world modeling and imitation learning (e.g. between loss and optimal model size). 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 so far best understood in 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), works studied the precise trade-off between model and dataset size (Hoffmann et al., 2022), as well as considering inference compute (Sardana & Frankle, 2023), repeated epochs (Muennighoff et al., 2024), parameter counting (Pearce & Song, 2024) etc. (Section 2.1).

In comparison, less is understood about scaling in *embodied AI*. Recent high-impact works show 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 (WM) (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 a few model sizes, shown in terms of downstream agent performance, confirming the intuition that 'bigger is better' (Sartor & 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 performance accurately predicted.

This paper helps close this gap. Similar to the 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 reward- or 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.

- For the first time, we show 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).
- 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).

[*]Equal contribution [1]Microsoft Research (SD now at Meta). Correspondence to: Tim Pearce <.>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

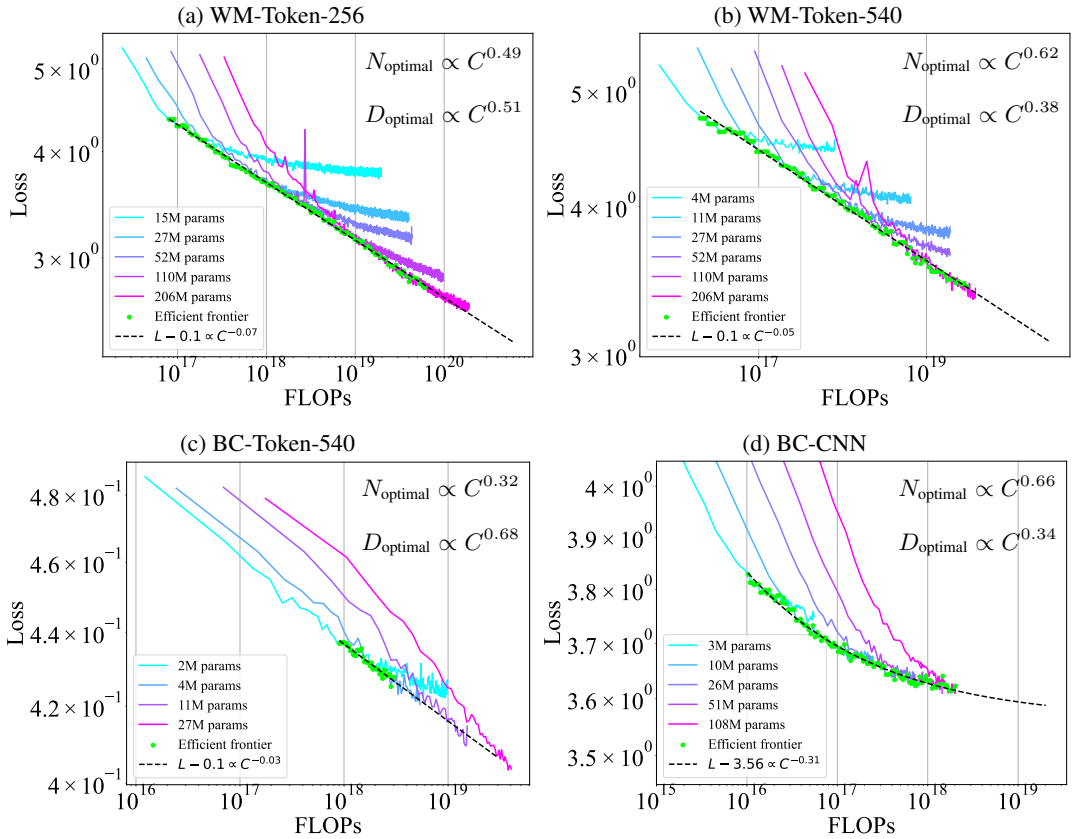

Figure 1. This paper observes that scaling laws, as originally found in LLMs, also emerge in the tasks of world modeling and BC, when studying the pre-training loss on large datasets of human behavior. (a, b) For world modeling, the power law coefficient determining optimal model size is affected by 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) Moving to a single continuous embedding per observation remedies this (BC-CNN), producing prototypical scaling laws and a more balanced optimal model size coefficient. Note BC-Token & BC-CNN losses are not directly comparable.

Organization. Section 2.1 provides detailed related work, contrasting the current understanding of scaling in embodied AI with other domains, and justifying pre-training loss as a proxy for online reward. Section 3 introduces details for our main experiments, including the architectures & datasets considered, and details of scaling laws analyses. Section 4 presents our main results in world modeling and BC. 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.

2. Background

This section introduces related work, and outlines arguments and evidence supporting using pre-training loss to study scaling in embodied AI.

2.1. Related Work

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 & Nunnari, 2016). The name derives from the *scale-invariant*[1] property of power laws. While early work suggested that power laws could be good empirical descriptors of pre-training loss in deep learning (Hestness et al., 2017; Rosenfeld et al., 2019), Kaplan et al. (2020) 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 pre-

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

cise analysis into the trade-off of model and dataset size, finding they should be increased in equal proportions. This conflicted with the suggestion that model size should be prioritized (Kaplan et al., 2020) – an incorrect conclusion that Pearce & Song (2024) showed largely arose from counting only non-embedding parameters.

Many other aspects of LLM scaling analyses are beginning to be refined. Su et al. (2024) revisited the methodology for finding scaling coefficients. Hägele et al. (2024) found that multiple independent cosine schedules could be reproduced more efficiently through a constant learning rate with multiple short decays, or stochastic weight averaging. Pearce & Song (2024) & Porian et al. (2024) found that well-tuned constant learning rates were sufficient to recover certain coefficients. 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 data regime. Sardana & Frankle (2023) factored in inference compute to the definition of compute-optimal. Isik et al. (2024) study the link between pre-training loss and downstream performance. A further line of research aims to explain *why* power laws are such a good descriptor of empirical deep learning (Hutter, 2021; Maloney et al., 2022; Bahri et al., 2024).

Scaling laws in image and video generation. Scaling laws have also been observed in auto-regressive modeling of video and images (Henighan et al., 2020; Tian et al., 2024). 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 different findings in the domain of world modeling – using updated methodologies to measure this trade-off, we find it *is* affected by the tokenizer.

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

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

Tuyls et al. (2023) offer a valuable initial investigation into scaling laws for BC. They fit power laws to both BC pre-training loss and online return, when scaling width of single-layer LSTM models on datasets generated by fixed high-reward policies. We extend this line of investigation by studying transformer models trained on datasets of human behavior, discovering effects of architecture choices on scaling coefficients. In addition we study scaling laws in world models for the first time.

2.2. Pre-training Loss as a Proxy for Performance

A major difference between scaling research in LLMs and embodied AI is that *LLM research uses pre-training loss* as the main variable of interest, while *embodied AI has focused on downstream online task performance*. This handicaps embodied AI scaling research – measuring online performance for a single model checkpoint in complex environments like robotics or modern video games is expensive in time and hardware, requiring multiple repeated runs to allow statistically significant comparisons. Furthermore, models may first require a period of fine-tuning before evaluation. By contrast, pre-training loss is available for free at any point of a model's training.

We believe embodied AI's focus stems from reports that validation loss is only weakly correlated with online performance (Hussenot et al., 2021; Li et al., 2024). However, such observations have been made with fixed-sized training datasets and held out validation sets, where effects of overfitting may be slightly beneficial. In contrast, scaling law studies are conducted in an **infinite data regime**, where datapoints are not trained on more than once, making train and test losses equivalent, and overfitting effects not applicable.

To evidence that pre-training loss can be a good proxy for online return in the infinite data regime, we conducted a meta-analysis of Tuyls et al. (2023), who were able to roll out a large number of checkpoints for two reasons. 1) They used simple lightweight environments (Atari & NetHack). 2) Their demonstration policy was high-skill, removing any need for fine-tuning. Figure 2 plots online environment return vs. pre-training loss for several environments (computed by tabulating pairs of points from Figure 6 & 10 in Tuyls et al. (2023)). The correlation coefficient for all environments is stronger than -0.94. Figure 3 shares evidence from our experiments that pre-training loss is well correlated with the video-generation quality of world models – providing correlation coefficients around 0.8. Further details in Appendix D.

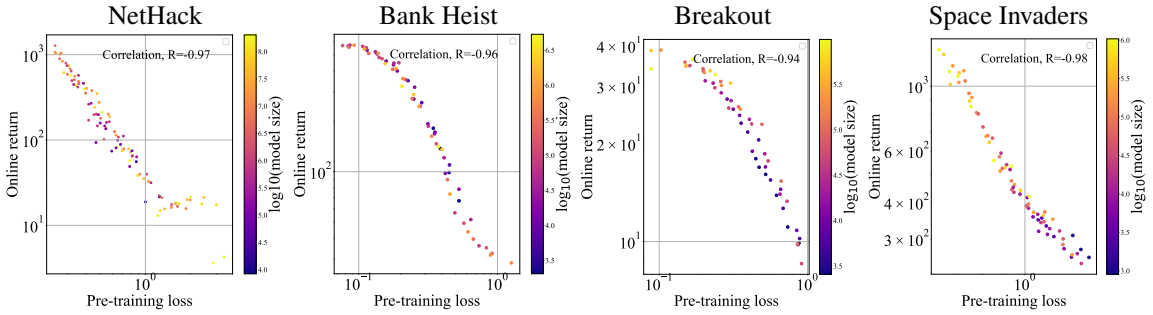

Figure 2. Our meta-analysis of Tuyls et al. (2023) evidences that pre-training loss is strongly correlated with reward in BC tasks when in the infinite data regime.

More intuitively, improving a next-token prediction loss in BC and WM requires models to 'know more' about behaviors and the environment, creating more useful pre-trained checkpoints for specialization to downstream tasks. In BC, better predicting the next action in a dataset of human behavior requires understanding the objectives humans are trying to complete, alternative social behaviors they might choose to perform, as well as making in-context inferences about the skill level and mental state of individuals. In WM, decreasing next-token prediction loss might follow a curriculum, first requiring a model to capture basic shapes and colors, then textures and physics, followed by rare object interactions, and finally even complex stochastic elements in the environment such as other intelligent agents.

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

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, $\mathbf{a}_t \in \{0,1\}^{d_a}$.

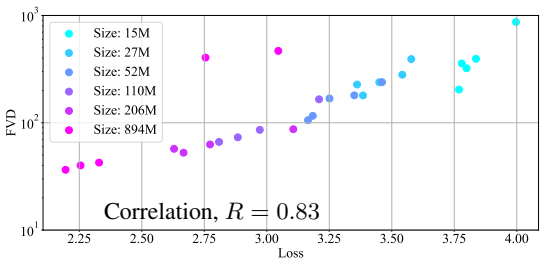

Figure 3. Our experiments suggest pre-training loss is a good proxy for generation quality of the world model, here FVD.

Given this data format, we consider two tasks. *World modeling* (WM) (Ha & 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,

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

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

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 4. 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,

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

where each item of the sequence is an integer within our vocabulary. A transformer is then trained to maximize the

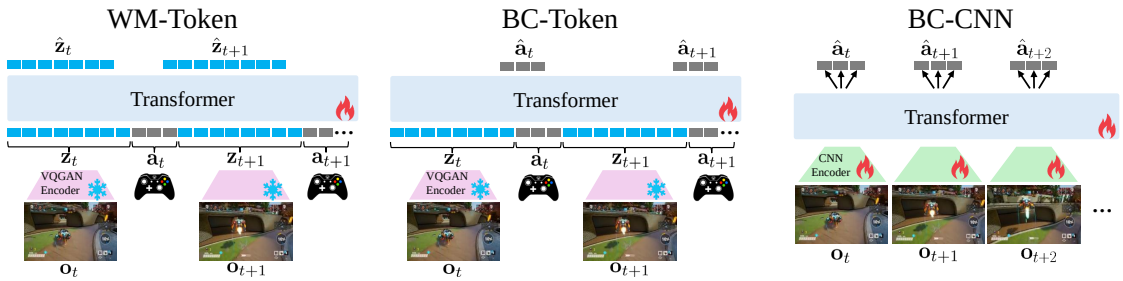

Figure 4. The World Modelling (WM) and Behavior Cloning (BC) tasks & architecture combinations considered in this work. The fire symbol signifies trainable components, the ice symbol signifies frozen pre-trained components.

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 A.

CNN architecture. Our second architecture differs in two ways. 1) Each image observation is input into the transformer as a single continuous embedding, extracted from a small trainable convolutional neural network (CNN). 2) Action dimensions are predicted independently (rather than in series), assuming $P(\mathbf{a}_t|\dots) \approx \prod_{i=1}^{d_a} P(a_t^i|\dots)$. One transformer forward pass produces an 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

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 – datasets should be large enough that training is done in the infinite data regime.
2. **Dataset diversity.** The behavior and environment must contain enough richness and variety that pre-training loss does not saturate across model sizes tested.

Many existing benchmark datasets fail to fulfill these criteria – either due to limited size, or because behavior is generated from a fixed policy, or the environment is too simple.

Our work primarily focuses on a dataset of human behavior collected in a video game *Bleeding Edge*. This is a fast-paced 4 vs 4 multiplayer game, with a range of characters, abilities and maps. Game play is highly complex due to cooperative and competitive dynamics. Success requires good high-level strategies (e.g. choosing which map regions to fight for), as well as fine-grained reactive control during combat. Figure 11 shows example dataset sequences.

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

As a secondary dataset we use RT-1 (Brohan et al., 2022), comprising 14 days of humans operating a robotic arm on a range of manipulation tasks such as 'pick banana from white bowl'. Using this smaller dataset allows us both to verify that conclusions on the large-scale video games dataset hold in a real-world robotics domain, and also allows us to run several small scale ablations in WM. Appendix C provides further details about the dataset and tokenizers used.

3.4. Scaling Analysis Methodology

We study the relationship between several quantities.

- 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 & 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, 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). Note training loss is an

accurate proxy for test loss when in the infinite data regime (Appendix A.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,

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

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

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

Scaling analysis. Scaling law analyses fit power law relationships predicting these compute-optimal quantities. For predicting optimal model and dataset size, we use,

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

with fitted constants a_0, a, b_0, b.[2] 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,

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

to the empirical training curves. In our implementation, we use SciPy's `curve_fit` function. We 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 & Song (2024),

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

[2]Note that by subscribing to $C = 6ND$ we find $a = 1 - b$; $N \propto C^a \implies C/D \propto C^a \implies D \propto C^{1-a}$. Hence, at times we only describe relationships in terms of $N \propto C^a$, with $N \propto D^{1-a}$ implied.

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).

	Frontier fit		Parametric fit	
Experiment	$N_{\text{optimal}} \propto C^a$	$D_{\text{optimal}} \propto C^b$	$N_{\text{optimal}} \propto C^a$	$D_{\text{optimal}} \propto C^b$
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

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 & 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 A.

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.

4.1. Scaling Analysis in World Modeling

Figures 5 & 6 present our results for the task of world modeling, with the scaling law coefficients summarized in Table 1. For WM-Token-256 we find that the optimal coefficients for model and dataset size are both ≈ 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.

Appendix Figure 15 visualizes the power law fits for the RT-1 robotics dataset, confirming that this predictable scaling behavior is not specific to human behavior in video games, and also emerges on real-world robotics tasks with high-skill human operators.

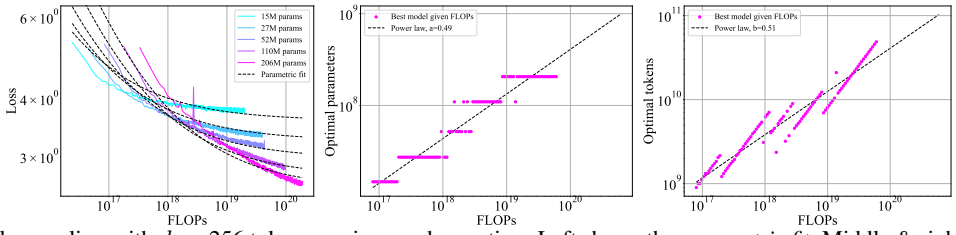

Figure 5. WM-Token scaling with d_z =256 tokens per image observation. Left shows the *parametric fit*. Middle & right show the *frontier fit* estimating optimal model & dataset size respectively.

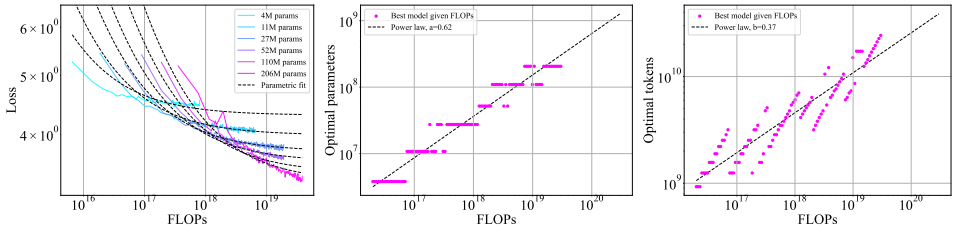

Figure 6. WM-Token scaling with d_z =540 tokens per image observation. Layout as in Figure 5. Compared to the results for WM-Token-256, the power law coefficient for N_{optimal} increases from 0.49 to 0.62.

4.2. Scaling Analysis in Behavior Cloning

We present our results on the scaling law coefficients for BC-Token in Figure 7. Despite sharing an architecture with WM-Token-540 we now observe the opposite dependence on model and dataset sizes. The coefficients skew heavily towards dataset size; $N_{\text{optimal}} = 0.32$, $D_{\text{optimal}} = 0.68$ (compared to $N_{\text{optimal}} = 0.62$, $D_{\text{optimal}} = 0.37$ – explained in Section 5.1). Furthermore, under the same compute budget the compute-optimal model sizes are significantly smaller. For a compute budget of 10^{18} and 10^{19} FLOPs we find that model sizes of 2M and 11M are compute-optimal for BC-Token-540 compared to 27M and 110M for WM-Token-540. In our experiments, we observe the losses for the BC-Token models take much longer to plateau leading to less overlap between model sizes. Hence, the *frontier fit* is not suitable for accurately estimating the scaling law coefficients, and we rely on the *parametric fit* for these results. Appendix E investigated whether smaller models would allow estimation via frontier fit, but this was unsuccessful.

To better understand the change in the scaling law coefficients, we now consider the BC-CNN architecture for the task of BC in Figure 8. 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

To test the extrapolation accuracy of our derived scaling laws, we train a $894M$ parameter WM-Token-256 model with an order of magnitude more compute than used for the scaling law analyses. Figure 9 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

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 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)

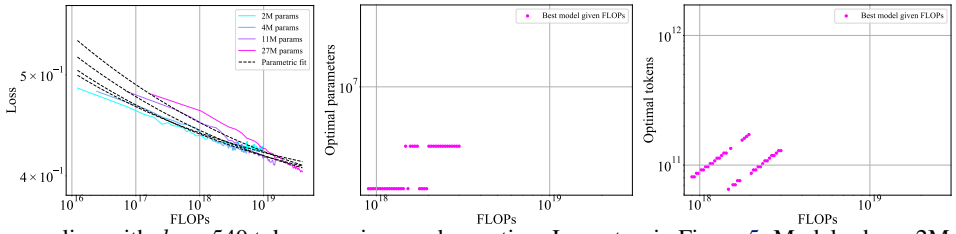

Figure 7. BC-Token scaling with $d_z = 540$ tokens per image observation. Layout as in Figure 5. Models above 2M parameters do not saturate over the FLOPs range considered and coefficients can not be inferred using the *frontier fit* method.

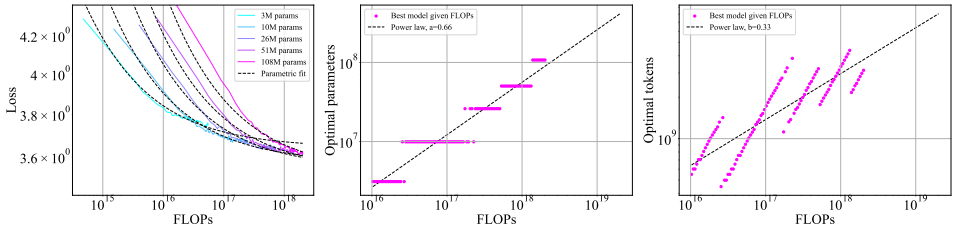

Figure 8. BC-CNN scaling. Layout as in Figure 5. Compared to the results for BC-Token the model sizes considered compute-optimal are considerably larger. The power law coefficient for N_{optimal} also increases from 0.32 to 0.66 skewing towards scaling model size as opposed to dataset size when scaling up compute.

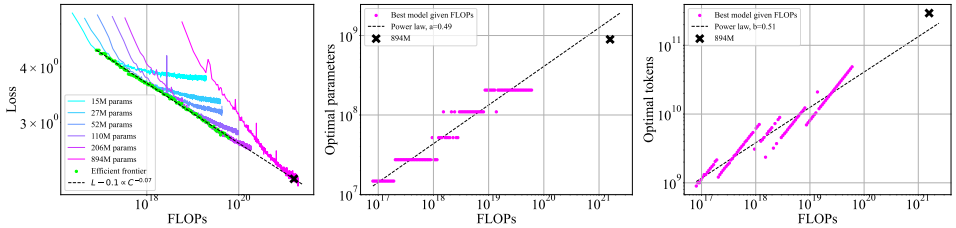

Figure 9. Testing the extrapolation capability of our derived scaling law for WM-Token-256 by training an 894M parameter model with an order of magnitude more compute than was used for the scaling law analyses. We observe good agreement between our predicted optimal loss/model size/number of training tokens (dotted lines) and our actual training run marked by 'x'.

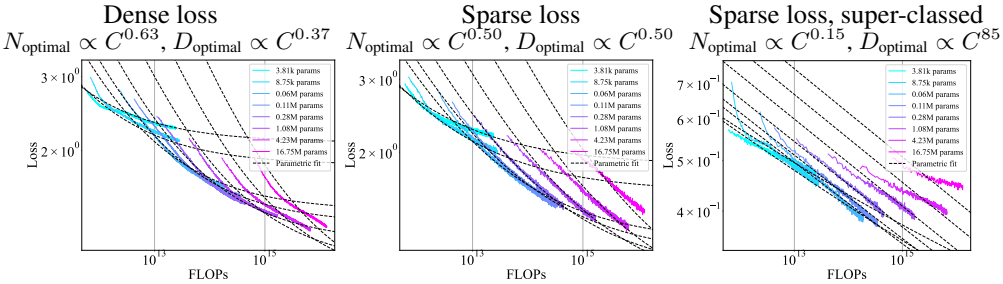

Figure 10. Training curves and parametric fit for character modeling experiments. The standard dense LLM loss has been modified to reflect properties of BC – a sparse loss (one of 16 tokens), and then additionally super-classing the targets into two classes.

5.1. 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 experi-

ments in language modeling. Transformers are trained on next-character prediction, on a dataset of Shakespeare text[3] 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 10 (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 (BC-CNN frontier fit) to 0.32 (BC-Token-540 parametric fit), indicating that the proposed mechanisms explain our findings.

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 4 that BC-Token receives $d_z + d_a = 556$ inputs for every action $\hat{\mathbf{a}}_t$ 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 ($556 \times 2 \times 1M \approx 1 \times 10^9$ FLOPs) than BC-CNN ($1 \times 2 \times 1M \approx 2 \times 10^6$ 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.

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.

Using the small scale RT-1 dataset, we conduct a more thorough investigation of the effect of tokens-per-image observation on scaling coefficients. First we train a range of image tokenizers with $z_o \in [16, 36, 64, 100, 256]$, visualized in Appendix Figure 14. 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, repeating three times.

Figure 16 plots all coefficient vs. tokens-per-image – we observe that the optimal parameter scaling coefficient increases with decreasing compression.

To investigate whether compression affects the optimal model size coefficient outside of embodied domains, 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.

Appendix B shows results. Under the character-level tokenizer (low compression), we find $N_{\text{optimal}} \propto C^{0.66}$. For the GPT-2 tokenizer (high compression), we find $N_{\text{optimal}} \propto C^{0.44}$. Hence, in language the more compressed representation also leads to a lower optimal model size coefficient.

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 pretraining of such models, we show that it is possible to recover scaling laws similar to those established for LLMs. Establishing this link is key to making efficient use of available resources in training compute-optimal models.

In world modeling, we show models can be smoothly scaled following best practices and insights from the LLM literature. We further establish that scaling coefficients are affected by the tokenizer's compression rate. In BC, the choice of architecture greatly impacts scaling coefficients. For 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.

[3]Shakespeare character dataset from: https://github.com/karpathy/nanoGPT

Impact statement

Limitations. While we show that scaling laws emerge in the infinite data regime for generative pre-training objectives in certain embodied datasets, future work could investigate scaling laws on alternative objectives and further datatsets. We have provided early evidence that this pre-training loss strongly correlates with online performance, but future work could seek further evidence, particularly in real-world robotic experiments.

Our observation that scaling laws arise in embodied AI suggests great potential in the direct scaling up of existing techniques. In consequence, it suggests a shift in focus towards data collection – how much data need be collected and with what strategies? Finally, we note our analysis has not factored in additional real-world constraints such as inference time.

Ethics. Data was provided via a partnership with the *Bleeding Edge* game developer *Ninja Theory*, who collected a large corpus of human gameplay data. Data collection was covered by an End User License Agreement (EULA) and our use of the data was governed by a data sharing agreement with the game studio, and approved with an IRB. This data was recorded between September 2020 and October 2022. To minimize risk to human subjects, personally identifiable information was removed from the data.

References

Bahri, Y., Dyer, E., Kaplan, J., Lee, J., and Sharma, U. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

Baker, B., Akkaya, I., Zhokov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Bakker, P., Kuniyoshi, Y., et al. Robot see, robot do: An overview of robot imitation. In *AISB96 Workshop on Learning in Robots and Animals*, volume 5, pp. 3–11, 1996.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Bousmalis, K., Vezzani, G., Rao, D., Devin, C., Lee, A. X., Bauza, M., Davchev, T., Zhou, Y., Gupta, A., Raju, A., et al. Robocat: A self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Cannavò, F. and Nunnari, G. On a possible unified scaling law for volcanic eruption durations. *Scientific reports*, 6 (1):22289, 2016.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Falcon, W. and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL https://github.com/Lightning-AI/lightning.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Hägele, A., Bakouch, E., Kosson, A., Allal, L. B., Von Werra, L., and Jaggi, M. Scaling laws and compute-optimal training beyond fixed training durations. *arXiv preprint arXiv:2405.18392*, 2024.

Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T. B., Dhariwal, P., Gray, S., et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Hilton, J., Tang, J., and Schulman, J. Scaling laws for single-agent reinforcement learning. *arXiv preprint arXiv:2301.13442*, 2023.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., and Corrado, G. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Hussenot, L., Andrychowicz, M., Vincent, D., Dadashi, R., Raichuk, A., Ramos, S., Momchev, N., Girgin, S., Marinier, R., Stafiniak, L., et al. Hyperparameter selection for imitation learning. In *International Conference on Machine Learning*, pp. 4511–4522. PMLR, 2021.

Hutter, M. Learning curve theory. *arXiv preprint arXiv:2102.04074*, 2021.

Isik, B., Ponomareva, N., Hazimeh, H., Paparas, D., Vassilvitskii, S., and Koyejo, S. Scaling laws for downstream task performance of large language models. *arXiv preprint arXiv:2402.04177*, 2024.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Karpathy, A. NanoGPT. https://github.com/karpathy/nanoGPT, 2022.

Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H. R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A ConvNet for the 2020s, March 2022. URL http://arxiv.org/abs/2201.03545. arXiv:2201.03545 [cs].

Maloney, A., Roberts, D. A., and Sully, J. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022.

Micheli, V., Alonso, E., and Fleuret, F. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Muennighoff, N., Rush, A., Barak, B., Le Scao, T., Tazi, N., Piktus, A., Pyysalo, S., Wolf, T., and Raffel, C. A. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

Neumann, O. and Gros, C. Scaling laws for a multi-agent reinforcement learning model. *arXiv preprint arXiv:2210.00849*, 2022.

Padalkar, A., Pooley, A., Jain, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Singh, A., Brohan, A., et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Pearce, T. and Song, J. Reconciling kaplan and chinchilla scaling laws. *TMLR*, 2024.

Porian, T., Wortsman, M., Jitsev, J., Schmidt, L., and Carmon, Y. Resolving discrepancies in compute-optimal scaling of language models. *arXiv preprint arXiv:2406.19146*, 2024.

Raad, M. A., Ahuja, A., Barros, C., Besse, F., Bolt, A., Bolton, A., Brownfield, B., Buttimore, G., Cant, M., Chakera, S., et al. Scaling instructable agents across many simulated worlds. *arXiv preprint arXiv:2404.10179*, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Rosenfeld, J. S., Rosenfeld, A., Belinkov, Y., and Shavit, N. A constructive prediction of the generalization error across scales. *arXiv preprint arXiv:1909.12673*, 2019.

Sardana, N. and Frankle, J. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *arXiv preprint arXiv:2401.00448*, 2023.

Sartor, S. and Thompson, N. Neural scaling laws for embodied ai. *arXiv preprint arXiv:2405.14005*, 2024.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Su, H., Tian, Z., Shen, X., and Cai, X. Unraveling the mystery of scaling laws: Part i. *arXiv preprint arXiv:2403.06563*, 2024.

Sutton, R. S. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

Tian, K., Jiang, Y., Yuan, Z., Peng, B., and Wang, L. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

Tuyls, J., Madeka, D., Torkkola, K., Foster, D., Narasimhan, K. R., and Kakade, S. M. Scaling laws for imitation learning in single-agent games. *arXiv preprint*, 2023.

Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., and Abbeel, P. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved vqgan. *ICLR*, 2022.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

The appendix is organized as follows.

- Appendix A contains details on the training of the model configurations, hyperparameters, and a description of the datasets used.
- Appendix B contains results from Section 5.3.
- Appendix C contains further details on training world models on robotics.
- Appendix D contains further results demonstrating the link between pre-training loss and performance.

A. Scaling Experiments Further Details

This section provides experimental details for all experiments on the primary Bleeding Edge dataset.

A.1. Hyperparameters

We trained two VQGANs from scratch with reconstruction losses.

- **BE-Small.** Based on (Esser et al., 2021), uses $d_z = 256$, $V_o = 4096$, $h = w = 128$, with 28M parameters, and a CNN design. It was trained on a single SkyGarden Bleeding Edge map.
- **BE-Large.** Based on (Yu et al., 2022), uses $d_z = 540$, $V_o = 4096$, $h = 180$, $w = 300$, with 150M parameters, and a vision transformer design. It was trained on all seven Bleeding Edge maps.

We selected the numbers of tokens per image based on qualitative assessment of reconstructions. We found that 256 tokens per image was the minimum that still allowed a reconstruction to capture the majority of salient gameplay details. However certain details still were lacking, such as an enemy player's health bars – hence we also considered a 540 token version that provided a higher quality reconstruction.

BC-CNN details. We use $h = w = 128$. The $0.6M$ paramter CNN is similar to that used by (Baker et al., 2022), however it uses ConvNext blocks (Liu et al., 2022). The CNN produces an embedding of size 1024 which is then put through a linear layer to obtain a vector matching the transformer's embedding dimension.

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.

N	**Layers**	**Num heads**	**Embed dim**
2M	3	3	180
4M	4	4	240
11M	6	6	360
15M	4	4	512
27M	8	8	512
52M	10	10	640
110M	15	12	768
206M	16	16	1024
894M	23	14	1792

A.2. Training Details

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

This section lists key hyperparameters. Note that it was important to find optimization settings that produced the lowest possible loss for a given model size. In general larger models require smaller learning rates. Our approach first optimized the smallest model through a grid sweep, we would then sequentially run a sweep over the next largest model, starting at the smaller model's optimized 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**	**Context length**	**Tokens per update**	**Learning rate**
15M	10	2,720	522,240	0.0007
27M	10	2,720	522,240	0.0007
52M	10	2,720	522,240	0.0007
110M	10	2,720	522,240	0.0007
206M	10	2,720	522,240	0.00057
894M	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**
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

Table 5. Hyperparameters for BC-Token with d_z =540 tokens per image observation.

N	**Seq len**	**Context length**	**Tokens per update**	**Learning rate**
2M	10	5,560	533,760	0.0005
4M	10	5,560	533,760	0.0005
11M	10	5,560	533,760	0.0001
27M	10	5,560	533,760	0.0001

Table 6. Hyperparameters for BC-CNN.

N	**Seq len**	**Context length**	**Items per update**	**Learning rate**
2M	10	10	2560	0.0003
3M	10	10	2560	0.0003
10M	10	10	2560	0.0003
26M	10	10	2560	0.0003
51M	10	10	2560	0.0003

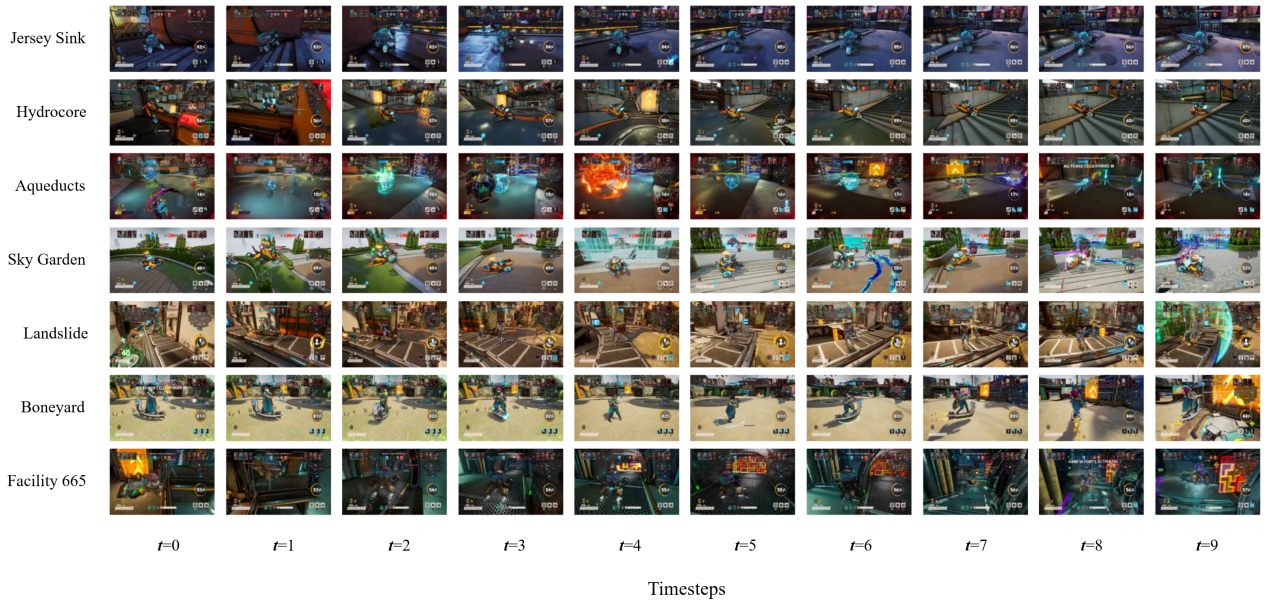

Figure 11. Example trajectories from a dataset of 8.6 years of human gameplay in the video game *Bleeding Edge* across 7 maps.

A.3. Dataset Details

Image observations were stored in MP4 format at 60fps, alongside binary files containing the associated 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.

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

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

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

A.3.1. Infinite data regime allowed FLOPs

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 $1.63\text{B} \times 556 = 906\text{B}$ 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 $C = 6ND = 6 \times 200\text{M} \times 3.6\text{T} = 4.3 \times 10^{21}$ FLOPs.

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

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

B. Further Analysis Details

Experimental results supporting Section 5.3.

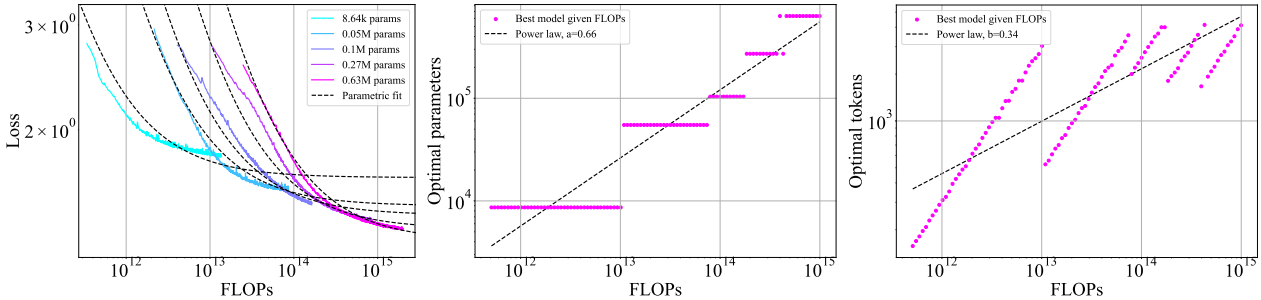

Figure 12. Relating to Section 5.3, character-level (low compression). Utilising the *frontier* fit (middle and right) we derive the power law coefficient for N_{optimal} as 0.66 and D_{optimal} as 0.34.

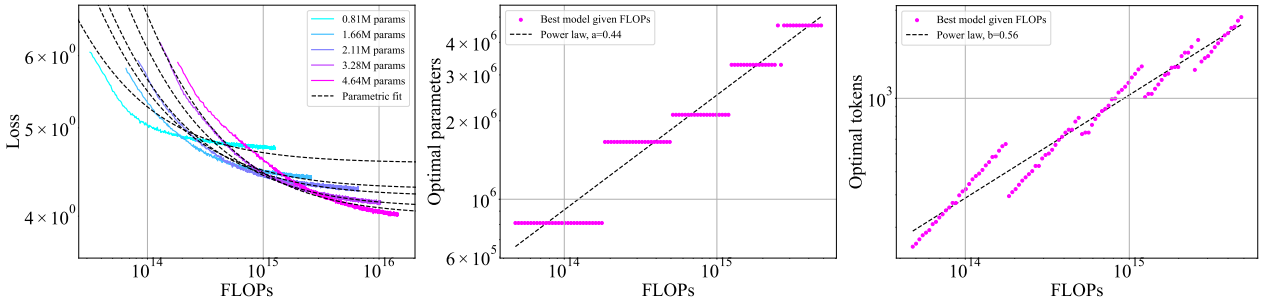

Figure 13. Relating to Section 5.3, GPT-2 tokenizer (high compression). Utilising the *frontier* fit (middle and right) we derive the power law coefficient for N_{optimal} as 0.44 and D_{optimal} as 0.56, an increase from 0.66 in Figure 12 found when utilising a lower compression character-level tokenizer.

C. World Modeling for Robotics Experimental Details

This section provides experimental details for WM experiments on the secondary RT-1 dataset.

C.1. Dataset

We resized the RT-1 dataset to 128x128 pixels per image. For action labels, we take the 3D `world_vector` coordinates, combined with the 1D `gripper_closedness_action` vector, to make an action vector with four dimensions. All are in the range -1 to 1, and these are discretized into 500 evenly spaced buckets.

C.2. VQVAEs

We trained a set of five VQVAEs using the implementation from https://github.com/nadavbh12/VQ-VAE. We set $z_o \in [16, 36, 64, 100, 256]$ and $V_o = 4096$, training each VQVAE for 40,000 updates on batches of 128. Reconstructions are visualized in Figure 14.

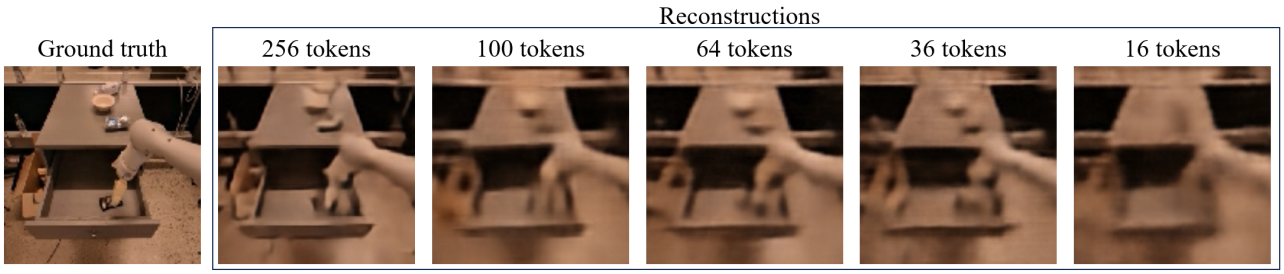

Figure 14. VQVAE reconstructions on the RT-1 dataset for differing numbers of tokens per observation, $z_o \in [16, 36, 64, 100, 256]$.

C.3. Transformer training details

Table 7 provides training details for the model sizes tested. Figure 15 shows one example set of training curves per VQVAE.

Table 7. Hyperparameters for WM-Token in RT-1 experiments.

N	**Seq len**	**Context length**	**Tokens per update**	**Learning rate**
0.08M	2	$2(z_o + 4)$	34,000	0.01
0.2M	2	$2(z_o + 4)$	34,000	0.005
0.28M	2	$2(z_o + 4)$	34,000	0.004
0.54M	2	$2(z_o + 4)$	34,000	0.0027
0.99M	2	$2(z_o + 4)$	34,000	0.002

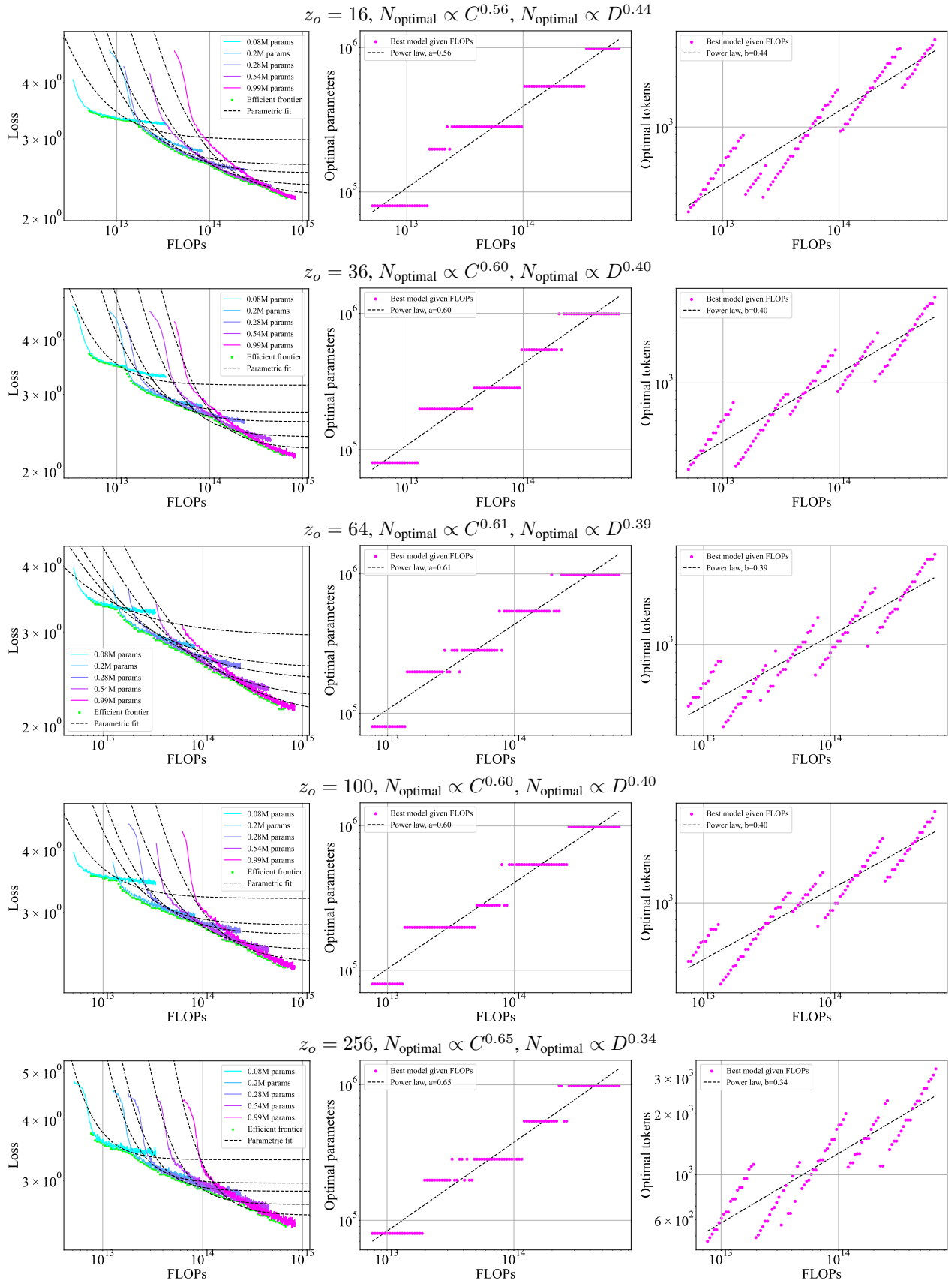

Figure 15. RT-1 experiments. Note that the optimal parameter coefficient increases with the number of tokens per observation.

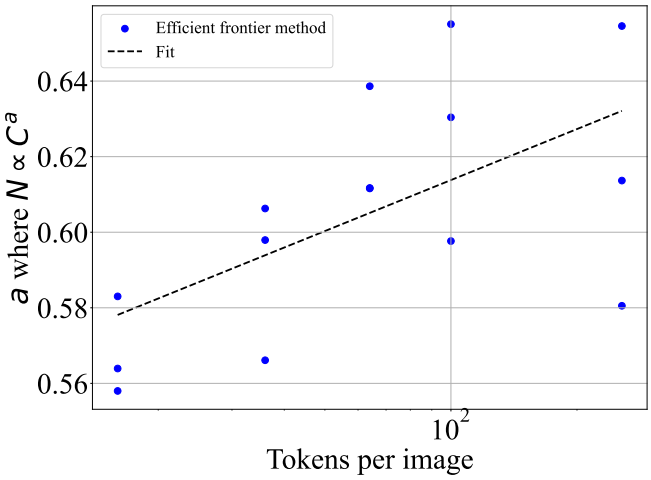

Figure 16. RT-1 experiments. Optimal parameter coefficient vs. number of tokens per observation, with three repeated runs per VQVAE. Correlation, $R = 0.61$.

D. Pre-training Loss vs. World Modeling Metrics

This section presents evidence for pre-training loss correlating with WM performance. 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 a world model, with the real sequence of observations, measuring FVD and LPIPS. 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 17 shows results. We find correlations of 0.77, 0.83 for LPIPS and FVD respectively. Two early checkpoints from the 894M model are the only significant anomalies to trend of metrics improving with loss. This evidences the strong relationship between pre-training loss and world model quality.

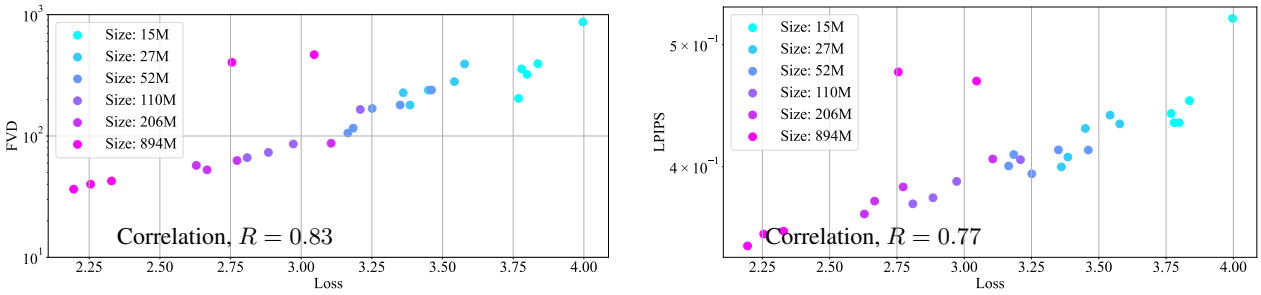

Figure 17. Our experiments suggest pre-training loss is a good proxy for generation quality of the world model, here FVD and LPIPS.

E. Additional Small Scale BC-Token Experiment

This section presents an additional set of results for a set of BC-Token models of smaller parameter counts. The aim had been to enable coefficient estimation using the frontier fit method. However, even at smaller sizes we did not observe the overlap required for this method – presented in Figure 18. Using the parametric fit method we observed, $N_{\text{optimal}} \propto C^{0.35}$, $N_{\text{optimal}} \propto D^{0.65}$, which are in line with the coefficients of larger BC-Token-540 models in Table 1.

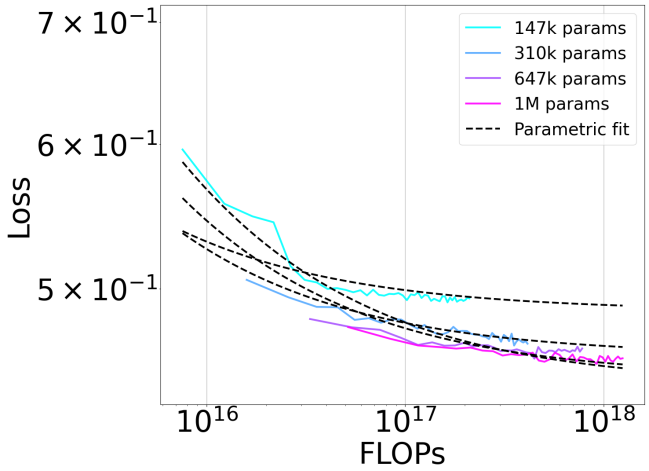

Figure 18. Additional set of results for a set of BC-Token models of smaller parameter counts. Using parametric fit method we observed, $N_{\text{optimal}} \propto C^{0.35}$, $N_{\text{optimal}} \propto D^{0.65}$.

Table 8. Transformer configurations for small scale BC-Token experiments.

N	**Layers**	**Num heads**	**Embed dim**
147k	1	1	25
310k	2	2	46
647k	2	2	86
1M	2	2	120

Table 9. Hyperparameters for BC-Token with d_z =540 tokens per image observation. Small-scale experiment.

N	**Seq len**	**Context length**	**Tokens per update**	**Learning rate**
147k	10	5,560	500,000	0.0005
310k	10	5,560	500,000	0.0005
647k	10	5,560	500,000	0.0005
1M	10	5,560	500,000	0.0005