

Seed2Scale: A Self-Evolving Data Engine with Parallel Worlds Expansion for Scalable Robot Learning

Cong Tai [†], Zhaoyu Zheng [†], Haixu Long [†], Hansheng Wu [†], Jialiang Wang, Zhengbin Long, Haodong Xiang, Shuyuan Wang, Rong Shi, Zhuo Cui, Shizhuang Zhang, He Wang, Biao Liu, Zhenzhe Sun, Tao Shen*

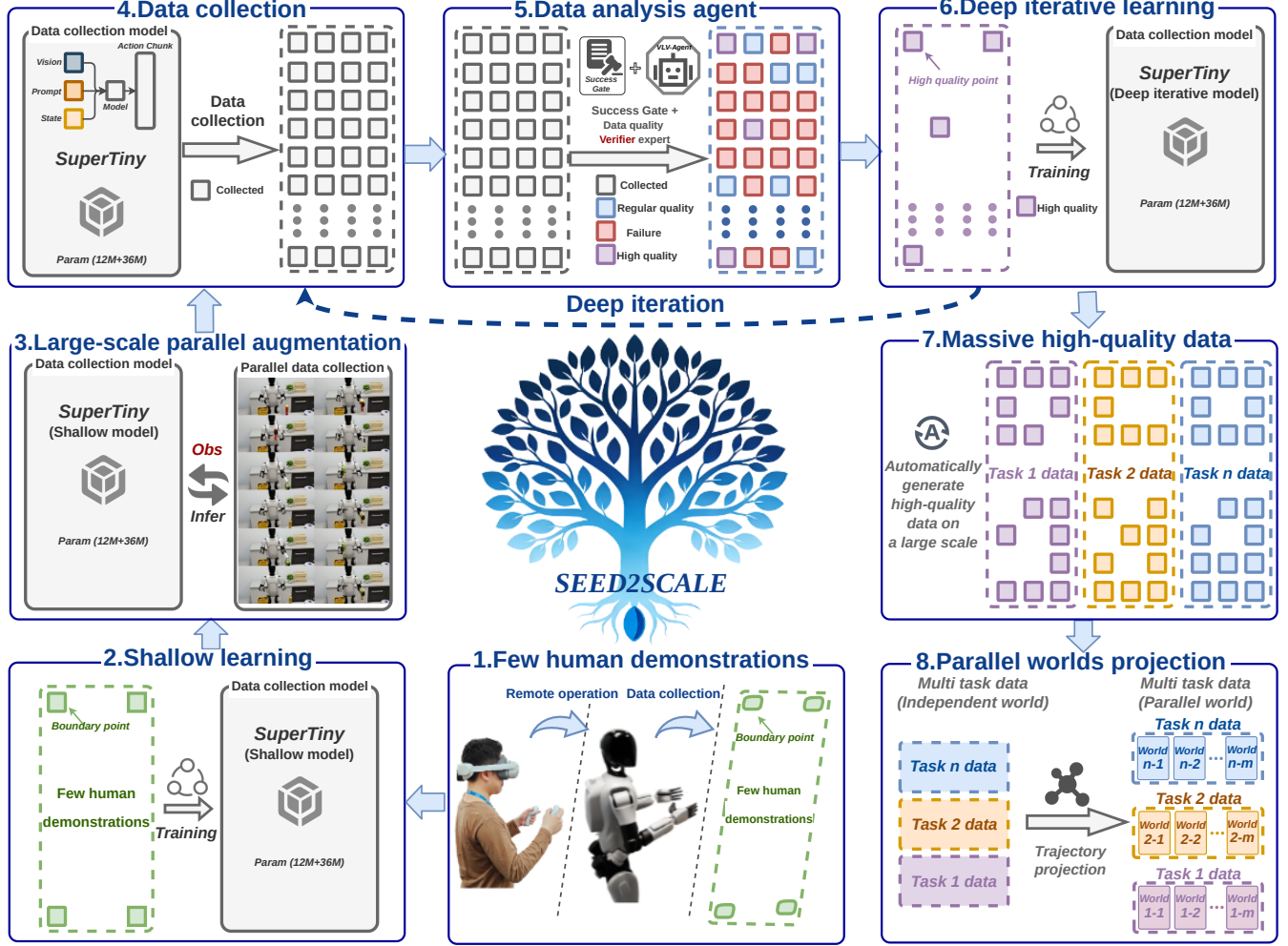


Fig. 1: The Seed2Scale self-evolving data engine. Starting with as few as four human demonstrations, the framework utilizes a lightweight SuperTiny VLA model, large-scale parallel data collection. A pretrained Vision-Language Model acts as a Verifier, autonomously scoring and filtering the raw trajectories to prevent model collapse. In the next stage, a parallel-world projection mechanism is introduced to achieve visual generalization on the basis of the position generalization accomplished through self-evolution, thereby yielding large-scale, high-quality, and diverse datasets for subsequent model training.

Abstract—Existing data generation methods for robot learning suffer from limited exploration, embodiment gaps, low signal-to-noise ratios, and domain shifts, leading to performance degradation during self-iteration and poor generalization to unseen scenes. To address these challenges, we propose Seed2Scale, a self-evolving data engine with parallel worlds expansion.

Starting with as few as four seed demonstrations, Seed2Scale first executes a self-evolution stage driven by a heterogeneous synergy of “small-model collection, large-model evaluation, and target-model learning”. Specifically, the lightweight Vision-Language-Action (VLA) model, SuperTiny, serves as a dedicated data collector for robust exploration. Concurrently, a pretrained Vision-Language Model (VLM) functions as a verifier to autonomously score and filter trajectories, supporting stable self-evolution in the evaluated tasks without performance collapse. Furthermore, Seed2Scale introduces a parallel worlds

[†] Equal contribution

* Corresponding author. Emails: shen.tao5@zte.com.cn

All authors are with ZTE Corporation, China.

stage, projecting self-evolved trajectories into different environments of the same task to generate more diverse data and enhance adaptability to unseen scenes, including real-world environments. Experimental results demonstrate that Seed2Scale exhibits significant scaling potential: as iterations progress, the success rate of the target model shows a consistent upward trend, significantly outperforming the seed baseline. Notably, Seed2Scale achieves a remarkable 75.38% success rate in zero-shot real-world evaluations, where baseline methods fail completely (0%). Code, datasets, assets, and model weights will be open-sourced.

I. INTRODUCTION

The emergence of Vision-Language-Action (VLA) models [1], [2], [3], [4], [5], [6] is accelerating the evolution of Embodied AI, integrating perception, reasoning, and control into a unified neural framework. However, the performance of these models remains strictly contingent upon large-scale, high-quality expert demonstration trajectories [7]. The onerous requirement for manual demonstrations results in a critical “data scarcity” bottleneck, fundamentally obstructing the scaling of robot learning.

Despite preliminary explorations into automated data generation, existing methods struggle to simultaneously achieve massive scale, high quality, and robust scene generalization. Data augmentation-based methods [8], [9], [10], [11] can expand expert trajectories through geometric perturbations. However, lacking active exploration, they are restricted to minor perturbations of human demonstrations, unable to generate novel action logic. Knowledge transfer from internet videos [12], [13] leverages massive visual data, but due to the inherent “embodiment gap”, it is difficult to precisely translate actions observed in videos into executable commands for physical robots. Critically, automated data collection suffers from an extremely low signal-to-noise ratio (SNR). Without effective quality evaluation, unsuccessful trials contaminate the training data, leading to cumulative performance degradation. Furthermore, even if spatial diversity is enriched within the source scene, transferring policies to unseen scenes, including real-world environments, remains challenging due to significant domain gaps.

To address these challenges, we propose Seed2Scale, a self-evolving data engine with parallel worlds expansion, designed to alleviate the data scarcity challenge in robot learning. Seed2Scale operates through a dual-stage pipeline: a primary self-evolution stage followed by a parallel worlds projection stage. In the self-evolution stage, Seed2Scale establishes a heterogeneous synergy architecture of “small-model collection, large-model evaluation, and target-model learning” to capitalize on the asymmetric advantages of models at different scales. Specifically, we first develop SuperTiny, a lightweight VLA data collector. Leveraging a streamlined query-based architecture and action chunking, SuperTiny achieves robust exploration with as few as four seed trajectories, avoiding the overfitting risks of large models in low-resource settings. Subsequently, the collector generates large-scale trajectories through parallelized rollouts and self-iteration. A pretrained Vision-Language Model [14], [15] is used as an automated verifier to provide quality scores

for filtering candidate trajectories during self-evolution. This mechanism averts model collapse from low-quality data, ensuring robust self-evolution. While the self-evolution stage primarily provides spatial generalization within the source scene, the parallel worlds stage projects self-evolved trajectories across different environments of the same task. By synthesizing diverse trajectories, Seed2Scale adapts the policy to unseen domains and real-world deployment.

To evaluate the effectiveness of the proposed method, we conducted extensive experiments across several representative embodied tasks [16] in both simulation and real-world environments. The experimental results demonstrate the superior performance and scaling potential of Seed2Scale: as the number of iterations increases, the success rate of the target model exhibits a robust consistent trend, improving significantly over the seed baseline, coupled with substantial success in zero-shot real-world evaluations where baseline methods fail completely.

In summary, the contributions of this work are fourfold:

- **Cost-Efficient Self-Evolving Engine:** Enables large-scale data generation from as few as four initial human demonstrations, significantly mitigating the reliance on manual data acquisition in robot learning.
- **VLM-Guided Data Curation Pipeline:** Employs a lightweight VLA model, SuperTiny, for robust data collection and a pretrained VLM as a verifier to filter low-quality trajectories, effectively preventing performance degradation during self-iteration.
- **Parallel Worlds Expansion Paradigm:** Projects evolved trajectories into diverse environments to improve policy performance on the reported unseen-scene evaluations.
- **Experimental Validation and Scaling:** Demonstrates robust performance scaling of the target model across iterations, significantly outperforming the seed baseline, and achieves a 75.38% success rate in the reported zero-shot real-world pick-and-place evaluation, without real-world fine-tuning.

II. RELATED WORK

A. Vision-Language-Action

Vision-Language-Action (VLA) models integrate visual-language understanding with robot action prediction. OpenVLA [17] autoregressively predicts discretized action tokens and demonstrates cross-task transfer. Recent work combines large-scale pretraining with continuous action generation: $\pi_{0.5}$ [4] uses a flow-matching action expert during post-training for cross-task and cross-embodiment generalization, while NVIDIA’s GR00T N1.6 [5] and Figure AI’s Helix [6] target scalable humanoid control. SmolVLA [1] uses a compact flow-matching architecture for resource-efficient, real-time deployment.

B. Automated Data Generation and Augmentation

To mitigate the prohibitive costs of human demonstrations, demonstration-based offline augmentation and retargeting expand expert datasets via geometric perturbations and inverse-

kinematics transformations [8], [11], [9], [10]. Recent methods diversify visual worlds through 3D scene editing and open-world synthesis while preserving executable actions [18]. Validated visual and spatial perturbations correlate with real-world VLA generalization, motivating our parallel-world rollout augmentation [19]. For contact-rich skills, physics-constrained synthesis combines VLM guidance with simulation to enforce contact-dynamics feasibility [20]. To shatter real-world data barriers, video-to-robot weakly supervised learning leverages massive internet videos via explicit cross-embodiment mapping [21], [22], [23], physics-grounded reconstruction [24], and scaling validations [25], and virtual-real simulation platforms for robotic task development and evaluation [26], while increasingly evolving toward implicit generative priors like discrete latent actions [12], inverse-dynamics world models [27], and video pretraining [28].

C. Vision-Language Models

VLMs have shown strong capabilities in visual understanding, language grounding, and video comprehension, making them promising tools for embodied AI. Recent models [14], [29], [30] further advance complex scene interpretation, long-horizon video understanding, and fine-grained spatial perception. These capabilities make VLMs well-suited for evaluation-centric roles in embodied AI, such as outcome assessment and trajectory quality evaluation, which motivates their use as automated verifiers in our pipeline.

III. SEED2SCALE

The synergy of Seed2Scale comprises three specialized model roles and one trajectory-level augmentation stage:

- **Small-Scale Collector** (π_{small}): A lightweight VLA enabling high-throughput parallel rollouts and seed-data bootstrapping.
- **Large-Scale Verifier** (Φ_{VLV}): A frozen Qwen3-VL [14], [15] that scores and filters candidate trajectories without human intervention.
- **Parallel Worlds**: A trajectory-conditioned stage that projects final self-evolved trajectories into Parallel Worlds, retargets their actions, and re-collects quality-filtered rollouts for final generalization.
- **Target Model** (π_{target}): SmolVLA [1], trained on the curated dataset $\mathcal{D}_{final}$ to distill verified visuomotor behavior.

By orchestrating these heterogeneous components, the Seed2Scale engine transforms a sparse set of 4 human demonstrations into a continuous flow of high-quality, verified training data, thereby bridging the gap between minimal expert input and generalized embodied intelligence. The workflow of the proposed framework, from seed bootstrapping to target model scaling, is illustrated in Fig. 1.

A. SuperTiny: Small-Scale VLA Collector

For high-throughput data collection, we develop *SuperTiny*, a lightweight VLA model for efficient inference and low-resource bootstrapping. As shown in Fig. 2, a heterogeneous encoding strategy unifies vision, language, and robot

state. A ResNet-18 backbone extracts visual features, projected via a 1×1 convolution. A pretrained T5-Small encodes language instructions. A compact MLP embeds robot state. All modalities are concatenated into a conditioning memory:

$$\mathcal{M}_t = [z_{vis}; z_{lang}; z_{state}] \quad (1)$$

A lightweight Transformer decoder processes learnable query embeddings Q via cross-attention over $\mathcal{M}_t$, and the decoder output is then projected through an action head to predict action chunks:

$$A_t = \text{ActionHead}(\text{Decoder}(Q, \mathcal{M}_t)) \in \mathbb{R}^{K \times d_a} \quad (2)$$

where K is the chunk size, and d_a is the action dimensionality.

To ensure smooth control during rollouts, we apply exponential temporal ensembling over overlapping action chunks:

$$a_t = \sum_{k=0}^{K-1} w_k \cdot A_{t-k}[k], \quad w_k = \frac{\exp(-\lambda k)}{\sum_{j=0}^{K-1} \exp(-\lambda j)} \quad (3)$$

where $A_{t-k}[k]$ is the k -th action in the chunk predicted at time $t-k$, and λ controls the decay rate.

B. VLM-as-a-Verifier

The primary challenge in automated data collection is the low SNR of raw trajectories. Without effective quality evaluation, failed explorations contaminate the training data and trigger performance degradation during self-iteration. To address this, we adopt a two-stage verification pipeline. First, a rule-based success gate filters out failed trajectories, ensuring by design that they do not enter the training set. The remaining trajectories are then scored for execution quality by a frozen, pretrained VLM (Qwen3-VL [14], [15]), which serves as a Vision-Language Verifier (VLV) to score and filter generated trajectories.

For each successful trajectory $\tau \in \mathcal{T}_{succ}$, the VLV receives a multimodal prompt consisting of: (1) the task instruction g , (2) the rollout video V_τ of the current attempt, and (3) a reference video V_{ref} demonstrating successful execution from the seed data. The verifier outputs a quality score:

$$S_\tau = \Phi_{VLV}(V_\tau, V_{ref}, g) \in [0, 10] \quad (4)$$

where higher scores indicate better task completion quality.

Based on these scores, only trajectories exceeding a quality threshold γ are admitted into the curated dataset $\mathcal{D}_{silver}$. The threshold γ was fixed a priori according to a manually designed 0–10 quality rubric. It was not optimized using the downstream test performance or evaluation scenes. This selective filtering restricts target-model training to trajectories that satisfy the VLV’s predefined quality criterion.

C. Seed2Scale Bootstrapping

To minimize reliance on expensive human demonstrations, For each evaluated task, Seed2Scale is initialized with four expert trajectories $\mathcal{D}_{seed}$ collected from four boundary or corner configurations, which correspond to the four corner

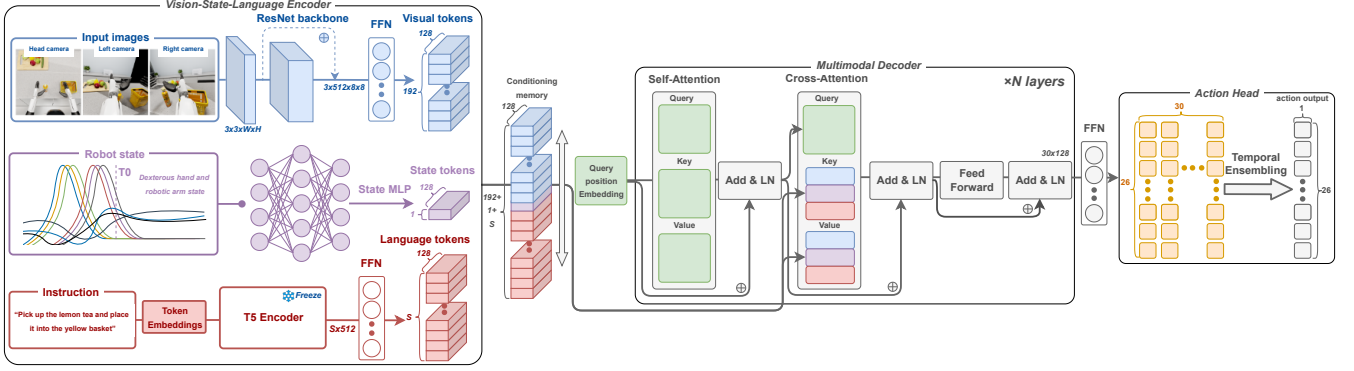


Fig. 2: Architecture of the SuperTiny VLA. The model integrates vision (ResNet), language (T5), and robot state encodings into a unified conditioning memory $\mathcal{M}_t$, which is processed by a lightweight Transformer decoder via cross-attention to predict action chunks. Temporal ensembling (Eq. 3) ensures smooth and consistent control.

positions of the tabletop workspace. Seed2Scale comprises two main stages: (1) self-evolution, a recursive data expansion loop that broadens the exploration frontier beyond the seed distribution for spatial generalization; (2) Parallel Worlds, which collect data via action projection in diverse environments for trajectory-based visual generalization, facilitating VLA transfer to unseen and real-world scenarios.

Self-evolution:

At iteration i , the lightweight collector $\pi_{small}^{(i)}$ is trained on dataset $\mathcal{D}^{(i)}$ (initialized as $\mathcal{D}^{(0)} = \mathcal{D}_{seed}$), then deployed in N_{env} parallel environments to generate raw trajectories:

$$\mathcal{T}_{raw}^{(i)} = \{\tau_j \sim \pi_{small}^{(i)}(\cdot | I, g, s)\}_{j=1}^{N_{rollout}} \quad (5)$$

where each trajectory $\tau_j = \{(I_t, s_t, a_t, g)\}_{t=0}^{T_j}$ contains visual observations, robot states, actions, and task instructions.

The VLV scores each trajectory τ_j (Sec. III-B). Only high-quality samples exceeding threshold γ are retained:

$$\mathcal{D}_{silver}^{(i)} = \{\tau_j \in \mathcal{T}_{succ}^{(i)} \mid S_{\tau_j} \geq \gamma\} \quad (6)$$

The dataset is then augmented for the next iteration:

$$\mathcal{D}^{(i+1)} = \mathcal{D}^{(i)} \cup \mathcal{D}_{silver}^{(i)} \quad (7)$$

At this stage, we call this variant **Seed2Scale***, which includes only the Self-evolution stage and provides spatially generalized data in the source world.

Parallel Worlds:

After self-evolution, we denote the final source-world dataset by $\mathcal{D}_{se} = \mathcal{D}^{(I)}$, where I is the final iteration in the self-evolution stage. To improve visual generalization, each source trajectory is projected in multiple target worlds with different visual contexts. A target world is specified as:

$$\mathcal{C}_m = (\mathcal{A}_m, \mathcal{P}_F, \mathcal{B}_m, \mathcal{K}_m) \quad (8)$$

where $\mathcal{A}_m$, $\mathcal{P}_F$, $\mathcal{B}_m$, and $\mathcal{K}_m$ denote the visual assets, source-aligned object poses, background and appearance configuration, and camera configuration, respectively. The task-relevant objects in all target worlds are placed at the same poses as in the source world.

Given a source trajectory $\tau_F = \{(I_t^F, s_t^F, a_t, g)\}_{t=0}^T$, we project its control trajectory into the target world under source-aligned task geometry. This design follows the

general scene-aware trajectory transformation paradigm, in which a demonstrated trajectory is transferred to a target scene before execution [31]:

$$\mathcal{M}_m(\tau_F) = \begin{cases} \tau_F, & \tau_F \in \mathcal{F}(\mathcal{C}_m), \\ \emptyset, & \text{otherwise.} \end{cases} \quad (9)$$

where $\mathcal{M}_m$ denotes the source-to-target projection operator and $\mathcal{F}(\mathcal{C}_m)$ denotes the target-world feasibility check. Specifically, the check verifies workspace validity, kinematic validity, collision avoidance, contact consistency, and trajectory smoothness. If the projected trajectory is feasible, its original control sequence is executed in the target world to collect new observations and robot states, and the resulting trajectory is filtered by the VLV:

$$\mathcal{D}_{final} = \mathcal{D}_{se} \cup \bigcup_{\tau_F \in \mathcal{D}_{se}} \bigcup_{m=1}^M \mathcal{Q}_\gamma[\text{Rollout}_{\mathcal{C}_m}(\mathcal{M}_m(\tau_F))] \quad (10)$$

Throughout the remainder of the paper, Seed2Scale* denotes the Self-evolution-only variant, the Parallel Worlds stage contributes to trajectory-based visual generalization, and Seed2Scale denotes their complete sequential combination. By leveraging high inference throughput, the engine, jointly driven by the self-evolution and Parallel Worlds, synthesizes thousands of diverse trajectories without manual intervention, overcoming the traditional data scarcity limitation in Embodied AI.

D. Target Model Training

The target model π_{target} (SmolVLA [1]) is trained with Conditional Flow Matching to generate structured action sequences through iterative denoising, enabling robust policy learning from $\mathcal{D}_{final}$.

Given visual and linguistic features o_t extracted by the VLM backbone, the model learns a vector field v_θ that maps noisy actions $A_t^\sigma = \sigma A_t + (1 - \sigma)\epsilon$ to the target vector field $u_t = A_t - \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise and $\sigma \in [0, 1]$ is the flow step. The training objective is:

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{\sigma, \epsilon, A_t, o_t} [\|v_\theta(A_t^\sigma, o_t) - u_t\|_2^2] \quad (11)$$

To ensure temporally coherent action chunks $A_t = (a_t, \dots, a_{t+K})$, SmolVLA employs an Action Expert architecture that interleaves Cross-Attention (CA) with Self-Attention (SA):

$$\text{ActionExpert}(o_t) = \text{SA}(\text{CA}(Q_{\text{actions}}, K_{o_t}, V_{o_t})) \quad (12)$$

The CA layers attend to vision-language context o_t , while the SA layers model temporal dependencies within the action sequence. This design implicitly enforces smoothness without requiring explicit regularization terms, resulting in stable closed-loop control during deployment.

IV. EXPERIMENTS

We design experiments to answer the following questions: (1) Can Seed2Scale significantly improve VLA policy performance with minimal seed data? (2) How does our work perform in real-world scenarios? (3) Does the self-evolution mechanism exhibit consistent scaling performance across iterations? (4) How does the self-evolution method compare with existing data augmentation methods? (5) Can the lightweight SuperTiny collector achieve competitive performance while enabling efficient data generation? (6) Does the VLV effectively serve as a quality gatekeeper?

A. Experimental Setup

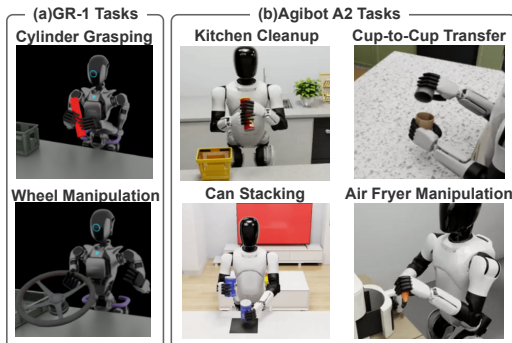


Fig. 3: (a) GR-1 robot tasks used for comparison with MimicGen and Seed2Scale*. (b) Agibot A2 robot tasks for Seed2Scale evaluation.

We evaluate Seed2Scale on the manipulation suite in Fig. 3, which comprises four Agibot A2 tasks on the RealMirror [16] platform and two GR-1 tasks against the MimicGen [8] baseline.

The Parallel Worlds consist of 10 distinct scenes for the A2 task suite, with variation in object instances, spatial arrangement, and background context, as illustrated in Fig. 4(a), some of which are constructed with 3D Gaussian Splatting (3DGS). Seen denotes the 10 scenes used for data generation, whereas Unseen denotes three additional disjoint scenes reserved for evaluation and excluded from training.

Each task is initialized with the original four seed demonstrations collected by teleoperation. In Parallel Worlds, the self-evolved trajectories are paired with regenerated scene-conditioned observations in the additional scenes. We use Success Rate (%) as the primary metric for model evaluation. Additional metrics are reported where applicable.

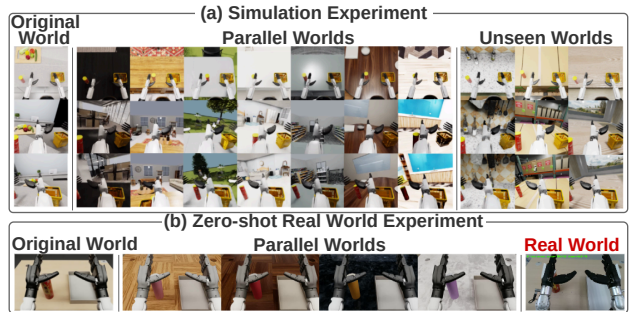


Fig. 4: (a) The original world, the parallel worlds projected from it, and the unseen worlds. (b) The worlds in the zero-shot real-world experiment.

The SuperTiny VLA collector contains 48M parameters with action chunk size $K = 30$, and the VLV is implemented using Qwen3-VL-32B [14]. The target model SmolVLA is trained for 100,000 steps using the AdamW optimizer ($lr = 3 \times 10^{-4}$, batch size 16).

B. Simulation Performance Improvement and Parallel Worlds Generalization

We first investigate whether Seed2Scale can significantly improve downstream policy performance from minimal seed data, and whether the Parallel Worlds expansion extends this generalization to unseen scenes. Table I reports the success rates on the four Agibot A2 tasks, evaluated separately on Seen and Unseen scenes.

The results reveal two distinct phases of capability acquisition. First, the self-evolution pipeline alone (**Seed2Scale***) boosts the average success rate on Seen scenes from 22.18% (**SeedOnly**) to 68.57%, validating the scaling benefit of self-evolution. Notably, Can Stacking exhibits the most significant improvement (7.50% $\rightarrow$ 65.90%), as the original 4-seed dataset provides insufficient coverage for the complex spatial alignments and stable placement required for stacking. We also observe that joint training on multiple tasks yields higher performance than single task training (Fig. 5), demonstrating beneficial cross-task knowledge learning.

Second, both SeedOnly and Seed2Scale* fail completely on Unseen scenes (0% success rate), confirming that trajectory scaling confined to a single source world cannot bridge the visual domain gap. Incorporating Parallel Worlds (**Seed2Scale**) resolves this limitation: it preserves the high performance on Seen scenes (67.01%) while dramatically elevating the Unseen success rate from 0% to an average of 65.34%. This isolates the contribution of scene diversity from that of additional action supervision.

These results support a scene-coverage-driven explanation of generalization: by re-projecting self-evolved trajectories across heterogeneous environments, Seed2Scale decorrelates the policy from source-scene appearance. This yields robust competence in unseen domains without compromising in-distribution accuracy.

C. Zero-Shot Generalization in Real-World Scenes

To examine whether the scene diversity introduced by Parallel Worlds can improve transfer to a physical robot,

TABLE I: Success rate (%) of the target model on the four Agibot A2 tasks, evaluated on Seen and Unseen scenes over 396 independent trials, with best results per task in **bold**. For each variant, a single SmolVLA is trained on data combined from all four tasks and evaluated separately on each task.

Task	Evaluated on Seen Scenes			Evaluated on Unseen Scenes		
	SeedOnly	Seed2Scale*	Seed2Scale	SeedOnly	Seed2Scale*	Seed2Scale
Kitchen Cleanup	24.63%	71.43%	68.50%	0.00%	0.00%	75.51%
Cup-to-Cup Transfer	23.50%	64.14%	60.61%	0.00%	0.00%	61.62%
Can Stacking	7.50%	65.90%	67.70%	0.00%	0.00%	69.19%
Air Fryer Manipulation	33.08%	72.82%	71.21%	0.00%	0.00%	55.05%
Average	22.18%	68.57%	67.01%	0.00%	0.00%	65.34%

TABLE II: Zero-shot success rate on the Pick-and-Place task, evaluated over 65 independent trials in an unseen real-world scene. Seed2Scale (n scenes) denotes a configuration trained with parallel-world data from n of the 7 training scenes.

Configuration	Episodes	Frames	Success Rate
SeedOnly	4	720	0.00%
Seed2Scale*	142	19,748	0.00%
Seed2Scale (3 scenes)	432	60,039	27.69%
Seed2Scale (5 scenes)	720	100,065	73.85%
Seed2Scale (7 scenes)	1,716	225,217	75.38%

we deploy the policies trained exclusively in simulation. Crucially, no real-world data collection or fine-tuning is performed. The evaluation adopts a typical *Pick-and-Place* task, with 7 scenes for training data and the real scene for evaluation, as illustrated in Fig. 4(b).

Table II reports the zero-shot success rate of each configuration on the physical robot. The training data expand from 4 episodes and 720 frames for SeedOnly to 1,716 episodes and 225,217 frames for Seed2Scale trained with 7 scenes. In contrast to the baselines, Seed2Scale successfully transfers to the real world, with its success rate increasing monotonically alongside scene diversity: 27.69% with 3 scenes, 73.85% with 5 scenes, and 75.38% with 7 scenes. This striking trend—from 0% in the single source world to 75.38% across 7 parallel scenes—demonstrates that Parallel Worlds is the primary driver of zero-shot generalization to real-world environments. The smaller marginal gain from 5 to 7 scenes is consistent with the diminishing returns of additional scene diversity.

D. Scaling Performance of the Self-Evolution Mechanism Across Iterations

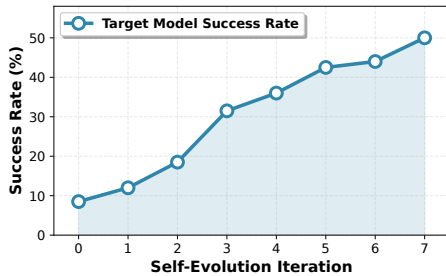


Fig. 5: Scaling performance of the target model (SmolVLA) across self-evolution iterations.

To investigate the scaling properties of the self-evolution,

TABLE III: Comparison with MimicGen on GR-1 tasks. We report the policy and replay success rates. Seed2Scale* outperforms MimicGen on both.

Metric	Task	MimicGen	Seed2Scale*
Policy Succ.	Cylinder Grasp	37.25%	66.00%
	Wheel Manip.	34.75%	93.25%
	Average	36.00%	79.63%
Replay Succ.	Cylinder Grasp	21.00%	86.96%
	Wheel Manip.	48.50%	67.86%
	Average	34.75%	77.41%

we track the target model’s performance on the most challenging Can Stacking task across 8 self-evolution iterations. Specifically, at each iteration i , we use the accumulated Can Stacking data generated by SuperTiny (up to iteration i) to train a SmolVLA model from scratch, then evaluate its success rate. This yields 8 independent single-task SmolVLA checkpoints, allowing us to observe how target model performance scales with the self-evolution process.

As shown in Fig. 5, starting from merely 4 seed demonstrations, the success rate follows a consistent upward trend as iterations progress, achieving substantial performance gains through the self-evolution mechanism.

E. Comparison with Existing Data Augmentation Methods

To validate the benefit of self-evolving exploration over conventional augmentation, we compare it with MimicGen[8] on GR-1 tasks. Both methods are evaluated under MimicGen’s experimental configuration for a fair comparison. As shown in Table III, self-evolution outperforms MimicGen on both metrics, achieving 86.96% replay success—a fourfold improvement—and downstream performance gains of 28.75% on Cylinder Grasping and 58.50% on Wheel Manipulation. MimicGen’s inverse-kinematics-based trajectories can be kinematically infeasible due to the absence of closed-loop environmental feedback. We further compare human demonstrations, MimicGen data, and self-evolved trajectories using three joint-motion metrics.

- **Total Variation (TV):** measures the cumulative magnitude of trajectory changes; lower values indicate smoother trajectories.
- **Mean Absolute Jerk:** Jerk represents the rate of change of acceleration. It measures motion naturalness.
- **High-Frequency (HF) Ratio:** FFT-based proportion of high-frequency energy relative to total non-DC energy.

TABLE IV: Quantitative comparison of trajectory quality. Lower values indicate better performance. Seed2Scale* produces trajectories with human-like smoothness, significantly outperforming MimicGen.

Metric	Expert Demonstration	MimicGen	Seed2Scale*
Total Variation	1.32	3.68	1.34
Mean Absolute Jerk	0.0063	0.0261	0.0047
HF Power Ratio (%)	0.22	2.07	0.30



Fig. 6: Comparison of robot action curves generated by expert demonstration, MimicGen, and the Seed2Scale*.

The experimental results are shown in Fig. 6. Trajectories generated by the self-evolution method closely match human demonstrations. Unlike MimicGen, which suffers from severe high-frequency jitter due to its IK-based transformations, our method produces smooth and stable motion curves that follow the expert profile.

The quantitative results in Table IV further confirm this improvement. MimicGen exhibits nearly $3\times$ higher Total Variation and $5.5\times$ higher Jerk, indicating substantial jitter and physical infeasibility caused by IK-based transformations. In comparison, our method achieves a Total Variation of 1.34 and an HF Ratio of 0.30%, both close to human demonstrations. Its Jerk is also lower than that of human data. Under these metrics, Seed2Scale* produces smoother joint motions than MimicGen, closely matching expert demonstrations while preserving task behavior.

F. Efficiency and Competitive Performance of the SuperTiny Collector

A core design choice in the self-evolution is the use of a lightweight VLA model as the data collector. To validate that SuperTiny achieves competitive task performance, we compare it against two representative VLA architectures: ACT [32] and Diffusion Policy [3].

Table V compares computational efficiency. SuperTiny achieves the fastest inference time (38.08ms) with a real-time control frequency of 26.3 Hz— $1.2\times$ faster than ACT and $3.6\times$ faster than Diffusion Policy. With only 48M parameters, SuperTiny is $5.5\times$ smaller than Diffusion Policy. This efficiency enables large-scale parallel data collection.

Fig. 7 compares the scaling performance of SuperTiny, ACT, and Diffusion Policy as data collectors. All models exhibit improved final performance, validating the effectiveness of our data selection and self-evolution mechanism. Notably, SuperTiny converges faster and achieves higher early success

TABLE V: Efficiency comparison of VLA models.

Model	Params	Inference Time	Frequency
ACT	52M	45.67ms	21.9 Hz
Diffusion Policy	265M	135.83ms	7.4 Hz
SuperTiny	48M	38.08ms	26.3 Hz

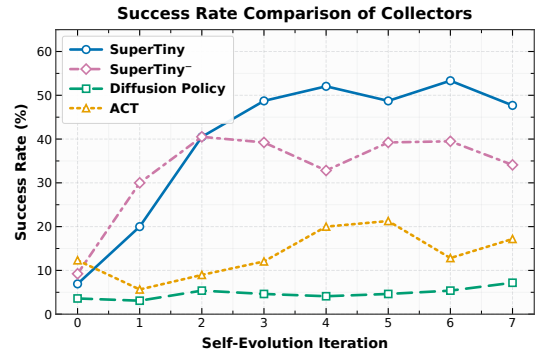


Fig. 7: Performance comparison of SuperTiny, ACT, and Diffusion Policy as data collectors across self-evolution iterations. SuperTiny⁻ denotes the variant without VLV-based quality filtering.

rates than the baselines, and reaches the highest final success rate, outperforming ACT and Diffusion Policy.

SuperTiny’s combination of real-time efficiency and strong task performance makes it the ideal collector for the self-evolution stage.

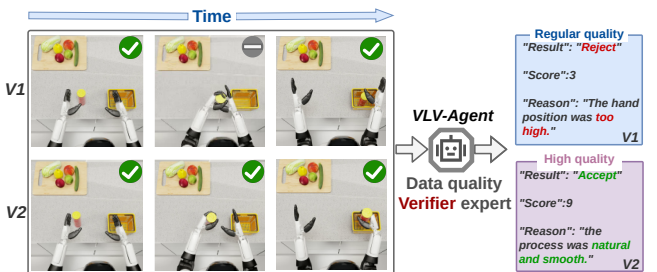


Fig. 8: VLV analysis of successful trajectories with different quality levels. Top: regular-quality execution (moderate score). Bottom: high-quality execution (highest score).

G. The VLV as Quality Gatekeeper

Beyond the rule-based Success Gate that ensures data validity by construction, we study the value of fine-grained quality gating among successful trajectories. As quantified in Fig. 7, the collector’s success rate remains below 15% in early iterations, and even among successful attempts, execution quality varies substantially. Comparing a baseline that retains all successful trajectories (SuperTiny⁻) against our strategy that filters suboptimal successes (SuperTiny), Fig. 7 reveals a significant performance gap. This ablation confirms that enforcing high trajectory quality—not just task completion—is essential for robust policy learning

Qualitative Visualization: As visualized in Fig. 8, the VLV effectively differentiates execution quality among successful trajectories: it assigns moderate scores to regular-quality executions (Top) and rewards high-quality, smooth completions

with the highest score (Bottom). This demonstrates human-aligned scoring that guides self-evolution.

V. CONCLUSIONS

In this paper, we presented Seed2Scale, a self-evolving data engine that addresses the data scarcity challenge in robot learning through a dual-stage paradigm combining self-evolution with parallel worlds expansion. By orchestrating a heterogeneous synergy among a lightweight collector, a pretrained verifier, and a target policy model, we show that the proposed pipeline can generate additional VLV-filtered trajectories from a small number of human demonstrations in the evaluated manipulation tasks. The Parallel Worlds expansion further decouples visual diversity from spatial exploration, facilitating transfer to unseen scenes. Experiments on the evaluated tasks show significant performance improvements across self-evolution iterations and successful deployment without real-world fine-tuning. In the future, we aim to extend Seed2Scale to long-horizon tasks and whole-body manipulation, while exploring tighter collector-verifier integration for more efficient and scalable self-evolution.

REFERENCES

- [1] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti, *et al.*, “Smolvla: A vision-language-action model for affordable and efficient robotics,” *arXiv preprint arXiv:2506.01844*, 2025.
- [2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, *et al.*, “Openvla: An open-source vision-language-action model,” in *Conference on Robot Learning*. PMLR, 2025, pp. 2679–2713.
- [3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, p. 02783649241273668, 2023.
- [4] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, *et al.*, “ $\pi 0.5$: a vision-language-action model with open-world generalization, 2025,” URL <https://arxiv.org/abs/2504.16054>, 2025.
- [5] NVIDIA GEAR Team, “GR00T N1.6: An improved open foundation model for generalist humanoid robots,” NVIDIA Research, Dec. 2025. [Online]. Available: [\url{https://research.nvidia.com/labs/gear/gr00t-n1.6/}](https://research.nvidia.com/labs/gear/gr00t-n1.6/)
- [6] Figure AI, “Helix: A vision-language-action model for generalist humanoid control,” February 2025. [Online]. Available: <https://www.figure.ai/news/helix>
- [7] O.-E. Collaboration, A. O’Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models,” *arXiv preprint arXiv:2310.08864*, vol. 1, no. 2, 2023.
- [8] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, “Mimicgen: A data generation system for scalable robot learning using human demonstrations,” *arXiv preprint arXiv:2310.17596*, 2023.
- [9] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. J. Fan, and Y. Zhu, “Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 16923–16930.
- [10] V. Pomponi, P. Franceschi, S. Baraldo, L. Roveda, O. Avram, L. M. Gambardella, and A. Valente, “Dynamimicgen: A data generation framework for robot learning of dynamic tasks,” *arXiv preprint arXiv:2511.16223*, 2025.
- [11] E. Ameperosa, J. A. Collins, M. Jain, and A. Garg, “Rocoda: Counterfactual data augmentation for data-efficient robot learning from demonstrations,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 13 250–13 256.
- [12] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, *et al.*, “Latent action pretraining from videos,” *arXiv preprint arXiv:2410.11758*, 2024.
- [13] Y. Wang, X. Li, W. Wang, J. Zhang, Y. Li, Y. Chen, X. Wang, and Z. Zhang, “Unified vision-language-action model,” *arXiv preprint arXiv:2506.19850*, 2025.
- [14] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, *et al.*, “Qwen3-vl technical report,” *arXiv preprint arXiv:2511.21631*, 2025.
- [15] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, *et al.*, “Qwen3 technical report,” *arXiv preprint arXiv:2505.09388*, 2025.
- [16] C. Tai, Z. Zheng, H. Long, H. Wu, H. Xiang, Z. Long, J. Xiong, R. Shi, S. Zhang, G. Qiu, *et al.*, “Realmirror: A comprehensive, open-source vision-language-action platform for embodied ai,” *arXiv preprint arXiv:2509.14687*, 2025.
- [17] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, “Openvla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [18] Y. Zhao, H. Fan, D. Chen, S. Chen, L. Chen, X. Li, G. Ren, and H. Dong, “Real2edit2real: Generating robotic demonstrations via a 3d control interface,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 23 106–23 116.
- [19] M. Sedlacek, P. Yefanov, G. Ponomatkin, J. Bardhan, S. Pilc, M. Fourmy, E. Kazakos, C. G. Snoek, J. Sivic, and V. Petrik, “Realm: A real-to-sim validated benchmark for generalization in robotic manipulation,” *IEEE Robotics and Automation Letters*, 2026.
- [20] Z. Jing, S. Yang, J. Ao, T. Xiao, Y.-G. Jiang, and C. Bai, “Humanoidgen: Data generation for bimanual dexterous manipulation via llm reasoning,” *arXiv preprint arXiv:2507.00833*, 2025.
- [21] S. Bahl, A. Gupta, and D. Pathak, “Human-to-robot imitation in the wild,” *arXiv preprint arXiv:2207.09450*, 2022.
- [22] E. Dessalene, P. Mantripragada, M. Maynord, and Y. Aloimonos, “Embodiswap for zero-shot robot imitation learning,” *arXiv preprint arXiv:2510.03706*, 2025.
- [23] Y. Liu, W. C. Shin, Y. Han, Z. Chen, H. Ravichandar, and D. Xu, “Immimic: Cross-domain imitation from human videos via mapping and interpolation,” *arXiv preprint arXiv:2509.10952*, 2025.
- [24] J. Hsieh, K.-H. Tu, K.-H. Hung, and T.-W. Ke, “Dexman: Learning bimanual dexterous manipulation from human and generated videos,” *arXiv preprint arXiv:2510.08475*, 2025.
- [25] R. Zheng, D. Niu, Y. Xie, J. Wang, M. Xu, Y. Jiang, F. Castañeda, F. Hu, Y. L. Tan, L. Fu, *et al.*, “Egoscale: Scaling dexterous manipulation with diverse egocentric human data,” *arXiv preprint arXiv:2602.16710*, 2026.
- [26] Z. Zheng, Z. Li, Y. Wang, Q. Xie, and L. Zeng, “Demonstrating dvs: Dynamic virtual-real simulation platform for mobile robotic tasks,” *arXiv preprint arXiv:2504.18944*, 2025.
- [27] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, *et al.*, “World action models are zero-shot policies,” *arXiv preprint arXiv:2602.15922*, 2026.
- [28] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, “Unleashing large-scale video generative pre-training for visual robot manipulation,” *arXiv preprint arXiv:2312.13139*, 2023.
- [29] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, *et al.*, “Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 24 185–24 198.
- [30] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, “Gpt-4o system card,” *arXiv preprint arXiv:2410.21276*, 2024.
- [31] N. Heppert, M. Argus, T. Welschhold, T. Brox, and A. Valada, “Ditto: Demonstration imitation by trajectory transformation,” in *2024 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2024, pp. 7565–7572.
- [32] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” *arXiv preprint arXiv:2304.13705*, 2023.