BrickFlow: Connectivity-Guided Brick Reconstruction

Peter Kulits[1,2] **Cordelia Schmid**[1]
[1]Inria, École Normale Supérieure, CNRS, PSL Research University
[2]Max Planck Institute for Intelligent Systems, Tübingen

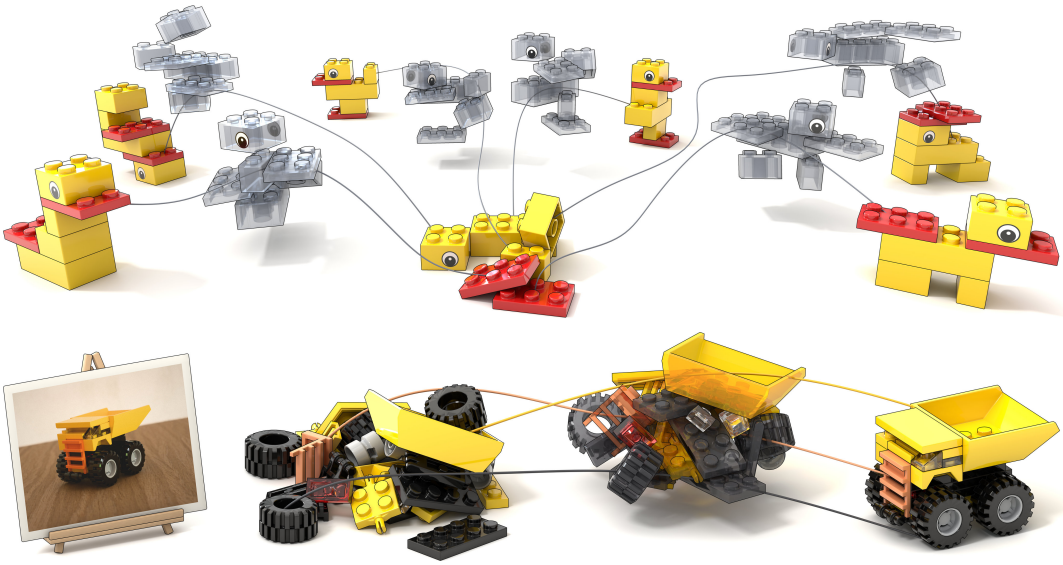


Figure 1: **Top:** We present BrickFlow, an SE(3) flow-matching model trained to take in a known set of LEGO pieces (pile) with noisy poses (translucent) and denoise them into some valid structure. **Bottom:** Using a procedural dataset 1% the size, we adapt it for image-conditioned reconstruction.

Abstract

Given a set of LEGO parts and an image of an assembled object, our task is to infer the pose of each part to assemble the object. In contrast with traditional 3D reconstruction, doing so requires satisfying discrete physical constraints, which provide structure but render the problem combinatorial and highly sensitive to small pose errors. To learn a prior over how parts fit together, we train a set-conditioned $SE(3)$ flow model on procedurally generated assemblies. We then finetune this model to inject image conditioning, using 1% as much data. However, while the learned flow captures global structure, it does not enforce connectivity. To address this, we introduce an analytic connectivity-field guidance to pull predictions toward connected assemblies at test time. Through this, we demonstrate improved build quality with model size, pretraining, and guidance. We make our models and data available for research purposes at `https://kulits.github.io/BrickFlow/`

1 Introduction

Imagine opening a new set of LEGO to find a box full of pieces, but no instructions. All you have is the picture on the front of the box. You can identify a number of parts visible on the surface, but the image only reveals the outside. The internal structure must be inferred: which bricks support which,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

what fills the spaces you cannot see. Every piece must end up with a precise position and orientation satisfying discrete connector-defined constraints. This is the problem we address: given an inventory of parts and a single image of the completed assembly, recover the 6D pose of each piece (see Fig. 1).

While sharing some of the ambiguities of traditional monocular 3D reconstruction – such as from occlusion – the task presents unique challenges. Though it is possible to computationally verify the validity of a given assembly, figuring out how to go from parts to structure requires understanding not just how pieces fit together, but how they fit together jointly, over possibly many at once. Contributing to this complexity, our assembly domain spans thousands of parts which connect in diverse ways.

We address this by learning a partset-conditioned generative assembly prior. To obtain sufficient supervision, we generate large-scale procedural structure data using a typed connector system [Kulits and Schmid, 2026] defined over LDraw [Jessiman, 1995–2026] parts, starting from randomly sampled pieces and iteratively placing new ones through compatible connectors. We use these to train an $SE(3)$ flow model that takes an unordered inventory together with noisy poses and removes noise.

The purpose of this stage is not to understand object semantics, but to learn the geometry of assembly: which parts can connect, how they tend to align, and how local connections compose into global structure. Rather than being human-designed or semantically meaningful, these assemblies provide constraint-rich samples from the physical grammar of LEGO assembly. Pretraining on this data amortizes search over connector-valid configurations into a reusable prior on assembly and structure.

We then adapt the assembly prior to perform partset-conditioned visual reconstruction. For this, we render a smaller set of procedural assemblies to create structure–image pairs. From the images, we extract features from a frozen visual encoder and finetune the flow model with injected cross-attention.

Although pretraining captures useful global assembly structure, learned predictions alone do not guarantee precise connectivity. Small translational and rotational errors can leave intended connections slightly misaligned. To address this, we introduce connectivity-field guidance, an analytic test-time energy defined over poses and connector geometry. The field is zero on exactly connected assemblies, and its gradient pulls predictions toward connected structure. Following diffusion posterior sampling (DPS) [Chung et al., 2023], we condition on connectedness with the field as the negative log-likelihood, correcting poses in late steps along its gradient at the model's predicted final assembly.

We evaluate our model on both inventory-conditioned assembly and image-conditioned reconstruction. Beyond pose accuracy, we evaluate assembly and connection validity. We demonstrate that assembly pretraining improves structural understanding, that task performance scales with model capacity, and that connectivity-field guidance improves the validity of assemblies when applied at inference time.

Our core contributions are as follows:

1. We train a set-conditioned $SE(3)$ flow model on procedurally generated LEGO assemblies to learn an adaptable assembly prior
2. We adapt this prior to single-image reconstruction using a fraction the amount of data
3. We introduce connectivity-field guidance, an analytic energy that injects known connector structure at test time, pulling predictions toward connected assemblies without retraining

2 Related Work

Brick Assembly. Prior work on brick assembly has addressed several specific subproblems. Legolization methods convert target geometry into brick layouts on discrete grids [Luo et al., 2015, Lennon et al., 2021]. BrickGPT [Pun et al., 2025] finetunes a language model to autoregressively place bricks, but predicts voxel positions over a small block vocabulary; LEGO-Maker [Ge et al., 2025] creates LEGO models from an image over a similarly restricted part set. For reconstruction, Tattersall et al. [2020] recover LEGO models from 24 calibrated views via space carving and cuboid fitting, but handle only four rectangular brick types; Wang et al. [2022] infer per-step brick poses from instruction manuals but address sequential single-part placement rather than joint assembly. BrickNet [Kulits and Schmid, 2026] introduce a typed connector system and graph-based generation over the full LDraw part library and finetune a language model to autoregressively produce brick structures from text. Each step places a part through a connection to the existing structure, so its outputs are connected by construction, and it takes no image or inventory. BrickBench [Kulits et al., 2026] evaluates coding agents on text-conditioned design of buildable assemblies with up to 4000 parts using the same library

and tests them under different constraints. Apart from the per-step placement of Wang et al. [2022], these methods choose which parts to use, and none is provided a fixed set of parts. Providing an inventory constrains the structure without determining it, as the same parts can build many assemblies. As it must be used exactly, a model cannot omit a part that does not fit and must connect every part. No prior work addresses inventory-conditioned assembly or reconstruction over the full part library.

3D Part Assembly. Generative part assembly methods predict rigid poses for a set of parts to form a complete object [Huang et al., 2020, Li et al., 2020], with recent work using score-based [Cheng et al., 2023], diffusion [Scarpellini et al., 2024, Zhao et al., 2025], hierarchical [Du et al., 2024], and flow-based [Li et al., 2025, Sun et al., 2025] formulations. GARF [Li et al., 2025] and Rectified Point Flow [Sun et al., 2025] are 3D-only, though Assembler [Zhao et al., 2025] is additionally image-conditioned. While these are architecturally similar (we employ a standard DiT [Peebles and Xie, 2023]), they differ in task, and are designed for arbitrary geometry encoded as sampled points.

SE(3) Flow Matching and Guided Sampling. The flow matching paradigm [Lipman et al., 2023, Liu et al., 2023] has been extended to $SE(3)$ for protein structure generation [Bose et al., 2024, Yim et al., 2023, Campbell et al., 2024], where, as in our setting, structures are built from a fixed vocabulary of units rather than arbitrary parts. Diffusion posterior sampling (DPS) [Chung et al., 2023] corrects each denoising step using gradients of a measurement likelihood; classifier [Dhariwal and Nichol, 2021] and classifier-free [Ho and Salimans, 2021] guidance follow the same principle with learned signals. Our connectivity-field guidance applies the DPS correction with an analytic energy over known connector geometry as the negative log-likelihood. No prior work integrates connectivity differentiably as test-time guidance in an assembly task.

3 Methods

3.1 Modeling

We represent LEGO assemblies using the LDraw [Jessiman, 1995–2026] modeling system, which consists of digital replicas of thousands of parts. In LDraw, a model is defined by a set of piece instances, each having type, material, and $SE(3)$ pose. It does not directly model part connectivity, for which we employ the BrickNet [Kulits and Schmid, 2026] typed connector system, an annotation layer of local connector frames over LDraw parts which defines how pieces connect to one another. These connection interfaces include stud–hole, hinge, axle, ball joint, and fixed connectors, each defining a family of allowed relative transforms between two parts. Stud-type connectors, for instance, allow for one degree of rotational freedom about an interface. Given an LDraw assembly, BrickNet parses posed parts into a graph where structure is defined by pieces as nodes and edges of connectivity. In this work, we use LDraw as the source of part geometry, while the BrickNet connector system is used to evaluate the validity of an assembly and integral to the production of procedural training data.

3.2 ProcBrick

To train our assembly prior, we introduce ProcBrick, a procedural generator of LEGO assemblies, and use it to synthesize a dataset of 100 million structures of 100 parts each. Parts are drawn from 6,339 geometry classes, which we use to group LDraw parts with identical colliders and connector interfaces. Starting with a random seed piece, we iteratively append pieces onto the structure. Rather than sample naively in space, we attempt placements at random compatible connector sites and reject those that collide. However, naive rejection sampling favors sparse assemblies: placements that extend into open space rarely collide, while those among existing parts, where further connections would form, do so more frequently. We instead stochastically accept collision-free placements to favor dense packing and, for each new part, numbers of connector joins and connected neighbors within the range of human-designed models, so that assemblies are compact and multiply connected. As sampling is autoregressive, each sequence prefix defines a valid structure and lengths can be varied in pretraining. For image-conditioned finetuning, we generate an additional one million ProcBrick structures, 1% as many as for pretraining, and render them with Blender. Rather than consistently include 100 parts, we balance counts between 2 and 100. For rendering, each part is assigned a printed or plain variant of its geometry class and a material, giving 12,663 distinct parts. We fit part and material frequencies to human-designed models (see Sec. H). See Fig. 2 for render samples. Although the procedural assemblies are not individually realistic as sets a human might construct, our intent is not to model semantics but to span the space of valid structure to achieve broad support.



Figure 2: **ProcBrick samples.** We provide sample renders from ProcBrick of increasing complexity. The observable differences in shading are due to random HDRI selection done at the time of rendering.

3.3 Problem

We formulate our task as 6-DoF pose prediction over an unordered multiset of known rigid parts:

$$\mathcal{P} = \{(p_i, m_i)\}_{i=1}^{N}, \tag{1}$$

where p_i denotes the part type and m_i is the material. The goal is to predict one transform per part,

$$X = \{T_i\}_{i=1}^{N}, \qquad T_i \in SE(3), \tag{2}$$

such that the transformed parts form the object depicted in a provided image I. Poses are predicted with the assembly centered at its mass-weighted centroid but in an arbitrary global orientation, so we evaluate up to a global rigid transform. As the inventory may contain duplicate parts, we do not fix ordering. Parts may have rotational symmetries. Training and evaluation therefore account for these three equivalences: global rigid transform, duplicate-part permutation, and intrinsic part symmetry.

We consider two settings. In inventory-conditioned assembly, which we refer also to as pretraining, the model receives only the part inventory and is tasked with predicting some valid assembly. In image-conditioned reconstruction, the model is provided an image and predicts the depicted assembly.

Pose distance. We measure distance between two poses of a part by how far material moves. With pose $T = (R, \mathbf{x})$ placing the part's center of mass at $\mathbf{x}$, the mean squared displacement of material is

$$d^2(T, T') = \|\mathbf{x} - \mathbf{x}'\|^2 + \operatorname{tr}\big((R - R')\,\Sigma\,(R - R')^\top\big), \tag{3}$$

where Σ is the part's second moment of mass per unit mass [Brégier et al., 2018]. Translation and rotation are thus measured in the same unit, LDraw units (LDU; 20 LDU is the width of a stud).

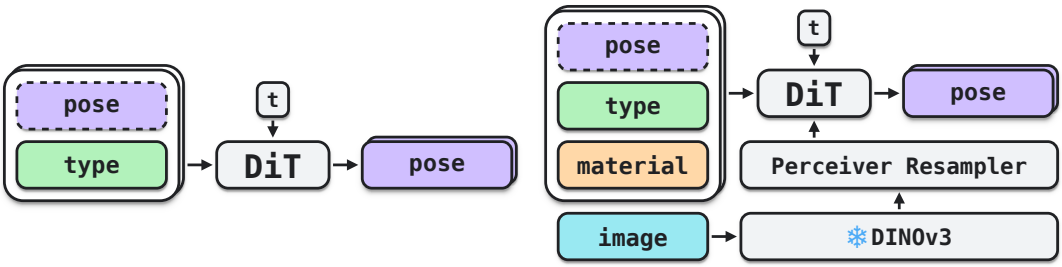
3.4 SE(3) Flow Matching

We train BrickFlow as a flow model over sets of rigid poses. Following flow-matching conventions, $t{=}0$ denotes the noise endpoint and $t{=}1$ the data endpoint. Let X_1 denote a target assembly and X_0 an initial noisy pose set. Noise translations are drawn from an isotropic Gaussian, shifted to zero mass-weighted centroid, with a standard deviation that depends only on the part count N and is fitted to the spread of N-part training assemblies; noise rotations are drawn uniformly from $SO(3)$. Because the input inventory is unordered and parts may be symmetric, we pair X_0 with X_1 by approximately minimizing the total mass movement $\sum_i \mu_i\, d^2(T_i^0, T_{\pi(i)}^1)$, with part masses μ_i, over permutations π of duplicate parts, part symmetries, and a global rotation [Klein et al., 2023]. For each training example, we sample t from a logit-normal distribution [Esser et al., 2024] and construct an interpolated pose set X_t between X_0 and X_1. Translations are then interpolated linearly,

$$\mathbf{x}_i(t) = (1 - t)\mathbf{x}_i^0 + t\mathbf{x}_i^1, \tag{4}$$

and rotations are interpolated along the geodesic on $SO(3)$,

$$R_i(t) = R_i^0 \exp\left(t \log\left((R_i^0)^\top R_i^1\right)\right). \tag{5}$$



(a) Structure-only pretraining (b) Image-conditioned finetuning

Figure 3: **Pipeline.** (a) BrickFlow denoises poses given known part types, conditioned on flow time t. (b) Finetuning adds materials and conditioning on frozen DINOv3 features with a Perceiver resampler.

These paths have constant velocities $\mathbf{v}_i = \mathbf{x}_i^1 - \mathbf{x}_i^0$ and $\boldsymbol{\omega}_i = \log\left((R_i^0)^\top R_i^1\right)^\vee$, the rotation vector of the relative rotation. The model predicts the translational and angular velocity of each part, $(\hat{\mathbf{v}}_i, \hat{\boldsymbol{\omega}}_i) = f_\theta(X_t, t, \mathcal{P}, I)$, where I and the materials are omitted during structure-only pretraining. We train it with the infinitesimal form of Eq. (3),

$$\mathcal{L} = \|\hat{\mathbf{v}}_i - \mathbf{v}_i\|^2 + w(t)\,(\hat{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i)^\top \tfrac{J_i}{\mu_i}\,(\hat{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i), \tag{6}$$

averaged over parts, where J_i is the part's inertia tensor and $w(t) = \left((1-t)/(1-\min(t, 0.9))\right)^2$ down-weights rotation errors near the data, as in FrameFlow [Yim et al., 2023]. The velocities imply a clean estimate $\hat{X}_1$, with $\hat{\mathbf{x}}_i^1 = \mathbf{x}_i(t) + (1-t)\hat{\mathbf{v}}_i$ and $\hat{R}_i^1 = R_i(t)\exp\left((1-t)\hat{\boldsymbol{\omega}}_i^\wedge\right)$.

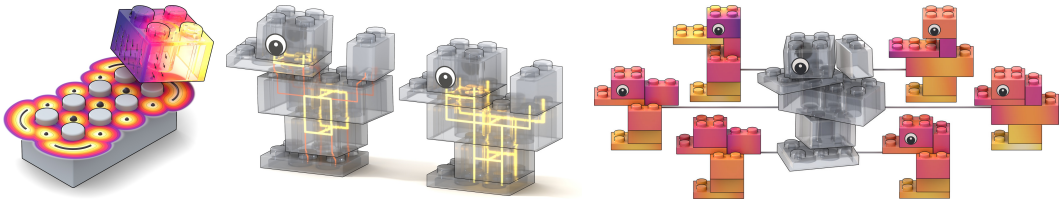
At inference, we integrate translations with 50 Heun steps [Karras et al., 2022]. Following FrameFlow and MultiFlow [Yim et al., 2023, Campbell et al., 2024], rotations instead advance a fraction $\kappa\,\Delta t$ of the geodesic toward $\hat{R}_i^1$ with a fixed rate κ=25, rather than the $\Delta t/(1-t)$ implied by the training interpolant, which roughly halves unguided validity (Sec. D). While the loss does not enforce that predicted parts connect, the model learns structure from seeing training samples that are connected.

3.5 Architecture

We use a standard diffusion-transformer (DiT) [Peebles and Xie, 2023] backbone without $SE(3)$-equivariant layers (Fig. 3). Prior work on $SE(3)$ flow matching for proteins [Yim et al., 2024] uses invariant point attention (IPA) layers from AlphaFold 2 [Jumper et al., 2021], but with data augmentation, non-equivariant models can match the data efficiency of equivariant ones given sufficient training [Brehmer et al., 2025]; AlphaFold 3 [Abramson et al., 2024] achieves strong results with a non-equivariant architecture and data augmentation, which we follow. Each input token represents a part: the token embedding is the sum of a learned part-type embedding, linear encodings of the current pose, and optionally a learned material embedding. As the input is unordered, no positional encoding is used. In the reconstruction experiment, we inject image conditioning through decoupled cross-attention [Ye et al., 2023] in each of the DiT blocks from part tokens to frozen DINOv3-L [Siméoni et al., 2025] features (CLS and patch tokens) from layer 18 of 24, compressed by a Perceiver resampler [Jaegle et al., 2021, Alayrac et al., 2022]. A keypoint head additionally predicts where each part's center of mass projects into the image, trained with a β-NLL loss [Seitzer et al., 2022]. We drop the image for 10% of training examples and sample with classifier-free guidance [Ho and Salimans, 2021] of strength 2, handing off to the pretrained model for the final steps ($t \geq 0.85$).

3.6 Connectivity Field

While large-scale pretraining teaches the model a useful global assembly prior, the learned flow does not explicitly enforce connector alignment. Small positional errors or drifts can leave two parts misaligned, leading to an assembly where the predicted structure might be qualitatively close, but disconnected. This is a natural failure mode for direct pose prediction in $SE(3)^N$: the set of connector-consistent configurations has measure zero in the full space. To address this, we introduce a connectivity-field guidance, a scalar energy defined over joint pose whose gradient nudges estimates toward connected configurations. We apply it at test time during sampling. The field encodes the same constraints that define a valid connection, but as a soft energy: enforcing them as hard constraints would presuppose the connection graph, which is precisely what is inferred jointly with the poses.



(a) Pair energy (b) Network (c) Energy minima

Figure 4: **Connectivity field.** (a) The energy E of a two-part assembly as a function of where the upper brick is placed on the lower one. It is zero (ink) exactly at valid seats ($c_h = 0$): isolated points where the brick is held rigidly, and arcs where it connects to a single stud and can swivel. The upper brick is colored by how far its material must move to reach a valid seat (Eq. (3)), with needles showing the direction. (b) E models each assembly as an electrical network: parts are nodes, and each candidate seat h adds a conductance g_h between two parts (drawn as wires, brighter for higher g_h). Seated connections join every part of the valid duck (right); the disordered estimate (left) consists of seated groups joined only through weak, unseated candidates, so its effective resistances Ω_{ab}, and hence E, are high. (c) A noisy estimate (center) and six nearby valid structures built from the same parts, colored by how far their material moves from the estimate. $E = 0$ on each of them.

Seats. For each candidate seat h, a pair of compatible connectors on parts a and b, we measure how far the parts are from a valid seat by $c_h \geq 0$, the mean squared material displacement (Eq. (3)) from their relative pose to the nearest allowed relative transform of the connector family, minimized over its free coordinates (e.g., stud rotation, axle slide). A perfect mate has $c_h = 0$.

Network. We treat the assembly as an electrical network with parts as nodes (Fig. 4b). Each candidate seat h contributes conductance $g_h = w_h/(\rho + q(c_h))$, where ρ is a small seat resistance, q is zero for c_h below a small tolerance and grows smoothly beyond it, and w_h tapers as connectors move apart (12–16 LDU). The candidate seats between two parts combine as $G_{ab} = (\sum_h g_h^4)^{1/4}$, so the best one dominates, and a weak conductance between every pair of parts keeps the network connected. Let Ω_{ab} denote the effective resistance [Doyle and Snell, 1984] between parts a and b. The energy is

$$E(X) = \frac{1}{N-1} \sum_{a<b} \phi\big(\Omega_{ab} - (N-1)\rho\big), \tag{7}$$

where ϕ is a smooth hinge that is zero for nonpositive arguments and positive otherwise (Sec. C).

Properties. If exactly seated connections join all parts, every pair is linked by a path of at most $N-1$ resistances ρ, so $\Omega_{ab} \leq (N-1)\rho$ and both E and its gradient vanish. Additional conductances can only lower Ω_{ab} (Rayleigh's monotonicity law [Doyle and Snell, 1984]), so loose contacts and redundant links do not disturb a connected assembly. In contrast, a detached group of parts has a high effective resistance to the rest and is pulled toward it through candidate seats in reach.

3.7 Test-Time Guidance

We seek samples from the energy-tilted distribution

$$\tilde{p}(X_1) \;\propto\; p(X_1)\, \exp\big(-\beta E(X_1)\big), \tag{8}$$

which reweights the model's output distribution toward connected assemblies. As $E \geq 0$, $\exp(-\beta E) \in (0, 1]$ for $\beta > 0$ is a valid likelihood of observing that the assembly is connected, so $\tilde{p}$ is the posterior given that observation. Following DPS [Chung et al., 2023], we approximate sampling from $\tilde{p}$ by evaluating the energy at the clean estimate $\hat{X}_1(X_t)$ and, for $t \geq t_{\text{start}}$, correcting the state:

$$X_{t+\Delta t} \;\leftarrow\; X_{t+\Delta t} \;-\; \gamma\, M^{-1} \nabla_{X_t} E\big(\hat{X}_1(X_t)\big), \tag{9}$$

where γ absorbs β, the correction is applied as a twist about each part's center of mass, and M weights each part's rotation by its squared radius of gyration. We clip each part's correction to 1 LDU and remove the mass-weighted mean translation, so the centroid stays fixed as in training. We use $\gamma = 2$ for all models and $t_{\text{start}} = 0.7$, or 0.85 for image-conditioned models, where the pretrained model takes the final steps; we select these on a subset of the ProcBrick evaluation set (Sec. D). On that subset, varying γ from 1 to 4, the clip from 0.5 to 2 LDU or removing it, or t_{start} from 0 to 0.8 changes validity by at most 2.1 points (Tab. 4). Each guided step adds one backward pass (Sec. F); no retraining or auxiliary networks are needed to apply the guidance at sampling time during inference.

Table 1: **Evaluation.** Procedural (ProcBrick) and human-designed (BrickNet) assemblies, 100 samples per structure. PT DiT-L receives only the inventory. Valid: 95% CI is shown in brackets.

Model	Guid.	Valid (%) ↑	Comp. ↓	Max comp. ↑	MM ↓	Coll. ↓
ProcBrick						
PT DiT-L	–	10.3 [9.9, 10.8]	7.0	33.5	72.6	18.6
	DPS	13.5 [12.9, 14.0]	2.9	43.0	72.5	17.8
IC DiT-S	–	6.5 [6.1, 6.9]	10.5	26.6	53.7	21.1
	DPS	8.9 [8.4, 9.4]	6.0	33.9	53.6	18.5
IC DiT-B	–	10.1 [9.6, 10.6]	8.4	29.9	48.4	17.6
	DPS	13.0 [12.5, 13.6]	4.7	36.7	48.4	15.3
IC DiT-L	–	13.4 [12.9, 14.0]	7.1	32.0	47.9	15.5
	DPS	15.8 [15.2, 16.4]	4.0	38.3	48.0	15.2
BrickNet						
PT DiT-L	–	25.5 [22.9, 28.0]	4.1	18.1	58.9	8.4
	DPS	31.7 [28.9, 34.4]	2.1	22.3	58.8	7.5
IC DiT-S	–	19.4 [16.8, 22.1]	6.3	15.2	34.9	11.3
	DPS	24.4 [21.4, 27.2]	3.9	18.6	34.8	9.7
IC DiT-B	–	26.1 [23.0, 29.1]	5.2	16.6	31.5	9.3
	DPS	31.7 [28.5, 34.8]	3.2	19.7	31.4	7.8
IC DiT-L	–	30.5 [27.4, 33.6]	4.4	17.8	31.0	8.0
	DPS	35.1 [32.0, 38.2]	2.8	20.6	30.9	7.3

4 Experiments

4.1 Setup

Models. We pretrain inventory-conditioned DiT-S/B/L (PT) on 100M procedural assemblies. For image-conditioned (IC) reconstruction, we finetune each on 1M rendered assemblies (Tab. 2). For ablations, we additionally train IC DiT-S from random initialization and ablate the amount of data.

Evaluation sets. We evaluate on 9,900 held-out ProcBrick assemblies, 100 for each part count from 2 to 100, and on the 512 models of the BrickNet [Kulits and Schmid, 2026] validation set, with 512×512 input images. We draw 100 samples. *DPS* uses connectivity-field guidance (Sec. 3.7).

Metrics. Pose error alone does not show whether an assembly holds together, so we report connectivity and collisions alongside it. *Valid*: BrickNet's connectivity and collision checks [Kulits and Schmid, 2026], applied to the whole assembly rather than per build step: the parsed connector graph has one component and no two parts collide by more than 1.1 LDU (↑); BrickNet's default of 1.0 LDU gives nearly identical results (Tab. 5). *Comp.* and *Max comp.*: number of connected components (↓) and size of the largest one (↑). *MM* (mass movement): root-mean-square material displacement (Eq. (3)) to the ground truth in LDU, with every part weighted equally, after optimal global alignment and symmetry-aware assignment of interchangeable parts (↓). *Coll.*: colliding part pairs per sample (↓). We report 95% confidence intervals from a bootstrap over structures.

4.2 Image-Conditioned Reconstruction

Tab. 1 summarizes reconstruction quality across both evaluation sets. Validity scales with model size: on BrickNet, unguided IC DiT-S/B/L reach 19.4%, 26.1%, and 30.5%, and guidance adds 4.6–5.6 points at every size. PT models scale likewise on BrickNet, from 13.6% to 25.5% (Tab. 5). Image conditioning reduces pose error, lowering MM from 58.9 to 31.0 LDU on BrickNet (PT DiT-L to IC DiT-L). ProcBrick follows the same trends at lower validity, as its assemblies are on average twice as large. Finetuning on 50% or 25% of the renders lowers the BrickNet validity of IC DiT-S only moderately, from 19.4% to 18.1% and 16.2% (Sec. G). Training IC DiT-S without the pretrained model, from random initialization and without the hand-off, lowers MM but increases the number of components and lowers BrickNet validity from 19.4% to 2.3% (Sec. G). Pretraining provides structural priors that improve connectivity even when individual part poses are slightly less accurate.

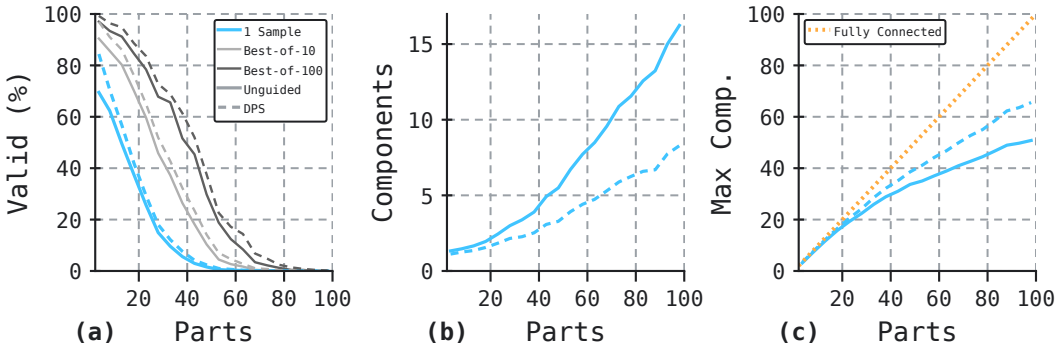


Figure 5: **Validity and connectivity across part counts.** IC DiT-L on ProcBrick, unguided (solid) and guided (dashed). (a) Validity of one sample and of the best of 10 and 100. (b) Connected components; guidance removes 16% of them for the smallest assemblies and about half beyond 80 parts. (c) Parts in the largest component, which grows with guidance from 51 to 66 for assemblies of 96–100 parts; the dotted line is a fully connected assembly.



Figure 6: **BrickNet reconstructions.** Outputs are selected from the guided IC DiT-L. As predictions are defined up to a global rigid transform, outputs are aligned to the ground truth as postprocessing.

4.3 Connectivity and Guidance

Without guidance, the number of components grows with part count (Fig. 5). Guidance reduces it at every part count, roughly halving it for the largest assemblies. Guidance changes MM by at most 0.3 LDU (Tab. 5). Guided samples also have fewer colliding pairs than unguided ones for every model (Tab. 1). Validity nonetheless falls with part count, with or without guidance, to below 1% beyond about 50 parts for single samples and beyond 85 parts for the best of 100. Large samples keep most parts in one component, but a single detached part or collision makes a sample invalid.

4.4 Qualitative Results

Reconstruction examples on BrickNet are shown in Fig. 6. The model recovers visible structure and infers plausible placements for occluded parts. Fig. 7 shows reconstructions from photographs of physical builds, given their part inventories.

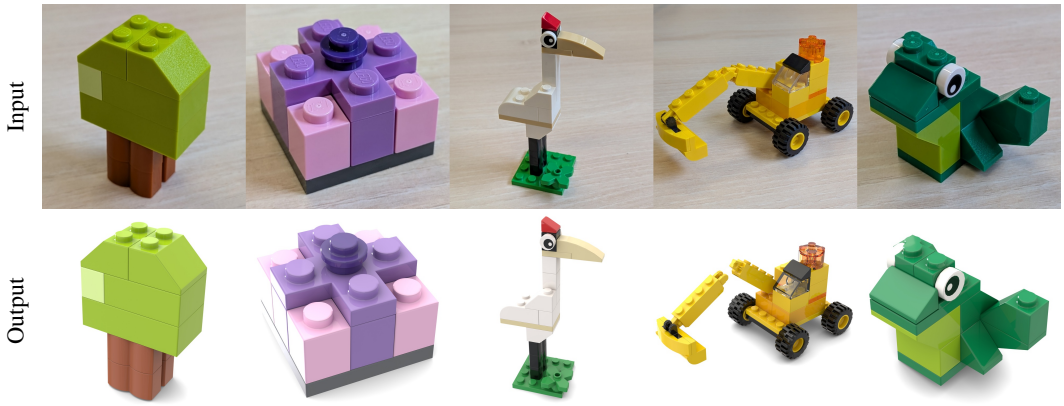


Figure 7: **Real-world reconstructions.** Inputs are photographs of physical builds. Outputs are from guided IC DiT-L, and are rendered following a postprocessing global alignment to set the camera.

5 Discussion and Limitations

While the connectivity-field guidance improves connectivity substantially, it does not guarantee fully connected or physically valid assemblies. The field has no explicit collision term, so it reduces collisions only indirectly, by pulling parts into seats. Achieving full validity may require better test-time optimization, for example with explicit collision terms or a discrete search over possibilities.

While our synthetic data appears to cover the test distribution fairly well, the model struggles in dense builds where parts look alike (Fig. 6). We suspect performance is limited by the distribution gap.

We assume that the inventory contains exactly the parts of the depicted assembly, and the method is not trained to handle missing, extra, or wrong parts. Future work might consider how to handle this, but should consider more strongly the much more interesting problem of jointly inferring the object's partset along with the poses of the parts. To make the method applicable to images captured in-the-wild, where the partset is not already known, such as to digitize a LEGO collection, interested readers might look to how co-generation is done in protein-generation literature such as in Campbell et al. [2024] with mixed continuous–discrete denoising.

Finally, while we train on assemblies of up to 100 parts, real-world LEGO sets can contain thousands. Scaling to larger assemblies will require addressing both computational cost and the increased ambiguity of inferring large hidden substructures from a single view.

6 Conclusion

Our work represents the first attempt to reconstruct a LEGO object given an image and its set of parts.

To achieve this, we first trained a set-conditioned $SE(3)$ flow model on procedurally generated assemblies (ProcBrick) to learn an adaptable assembly prior. Second, we adapted this prior to single-image reconstruction using a fraction of the data, and despite training only on renders, demonstrated that the resulting model can produce plausible reconstructions of physical builds from photographs. Finally, we introduced connectivity-field guidance, a test-time energy that pulls predictions toward connected assemblies during sampling, applicable without retraining, raising validity for every model at negligible pose cost and halving the number of disconnected components in large assemblies.

Acknowledgments and Disclosure of Funding

This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "France 2030" program, reference ANR-23-IACL-0008 (PR[AI]RIE-PSAI project). Cordelia Schmid would like to acknowledge the support by the Körber European Science Prize. We are grateful to Michael Black for his encouragement and support.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630:493–500, 2024.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. SE(3)-stochastic flow matching for protein backbone generation. In *International Conference on Learning Representations*, 2024.

Romain Brégier, Frédéric Devernay, Laetitia Leyrit, and James L. Crowley. Defining the pose of any 3D rigid object and an associated distance. *International Journal of Computer Vision*, 126(6):571–596, 2018.

Johann Brehmer, Sönke Behrends, Pim de Haan, and Taco Cohen. Does equivariance matter at scale? *Transactions on Machine Learning Research*, 2025.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=kQwSbv0BR4`.

Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. Score-PA: Score-based 3D part assembly. In *British Machine Vision Conference*, 2023.

Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 1984.

Bi'an Du, Xiang Gao, Wei Hu, and Renjie Liao. Generative 3D part assembly via part-whole-hierarchy message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20850–20859, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Jiahao Ge, Mingjun Zhou, Hanyou Zheng, Hao Xu, and Chi-Wing Fu. LEGO-Maker: Autoregressive image-conditioned LEGO model creation. *ACM Transactions on Graphics*, 44(6):201:1–201:15, 2025. doi: 10.1145/3763285.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

Jialei Huang, Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas Guibas, and Hao Dong. Generative 3D part assembly via dynamic graph learning. In *Advances in Neural Information Processing Systems*, 2020.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021.

James Jessiman. LDraw, 1995–2026. URL `https://ldraw.org/`.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. In *Advances in Neural Information Processing Systems*, 2023.

Peter Kulits and Cordelia Schmid. BrickNet: Graph-backed generative brick assembly. In *CVPR*, June 2026.

Peter Kulits, Yiqing Xu, R. Kenny Jones, Cordelia Schmid, and Jiajun Wu. BrickBench: Evaluating agentic brick design, 2026. URL https://arxiv.org/abs/2610.12452.

Kyle Lennon, Katharina Fransen, Alexander O'Brien, Yumeng Cao, Matthew Beveridge, Yamin Arefeen, Nikhil Singh, and Iddo Drori. Image2Lego: Customized LEGO set generation from images. *arXiv preprint arXiv:2108.08477*, 2021.

Sihang Li, Zeyu Jiang, Grace Chen, Chenyang Xu, Siqi Tan, Xue Wang, Irving Fang, Kristof Zyskowski, Shannon P. McPherron, Radu Iovita, Chen Feng, and Jing Zhang. GARF: Learning generalizable 3D reassembly for real-world fractures. In *ICCV*, 2025.

Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3D part assembly from a single image. In *European Conference on Computer Vision*, 2020.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. Legolization: Optimizing LEGO designs. *ACM Transactions on Graphics*, 34(6), 2015. doi: 10.1145/2816795.2818091.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, October 2023.

Ken Perlin. Improving noise. In *ACM SIGGRAPH*, pages 681–682, 2002.

Ava Pun, Kangle Deng, Ruixuan Liu, Deva Ramanan, Changliu Liu, and Jun-Yan Zhu. Generating physically stable and buildable brick structures from text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14798–14809, 2025.

Gianluca Scarpellini, Stefano Fiorini, Francesco Giuliari, Pietro Morerio, and Alessio Del Bue. DiffAssemble: A unified graph-diffusion model for 2D and 3D reassembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations*, 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Tao Sun, Liyuan Zhu, Shengyu Huang, Shuran Song, and Iro Armeni. Rectified point flow: Generic point cloud pose estimation. In *NeurIPS*, 2025.

George Tattersall, Dizhong Zhu, William A. P. Smith, Sebastian Deterding, and Patrik Huber. Reconstructing creative LEGO models. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Chin-Yi Cheng, and Jiajun Wu. Translating a visual LEGO manual to a machine-executable plan. In *ECCV*, pages 677–694, 2022.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, Jose Jimenez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noe. Fast protein backbone generation with SE(3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

Jason Yim, Andrew Campbell, Emile Mathieu, Andrew Y. K. Foong, Michael Gastegger, Jose Jimenez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Frank Noe, Regina Barzilay, and Tommi Jaakkola. Improved motif-scaffolding with SE(3) flow matching. *Transactions on Machine Learning Research*, 2024.

Wang Zhao, Yan-Pei Cao, Jiale Xu, Yuejiang Dong, and Ying Shan. Assembler: Scalable 3D part assembly via anchor point diffusion. In *SIGGRAPH Asia Conference Papers*, 2025. doi: 10.1145/3757377.3763972.

A Training Settings

Table 2: **Training settings.** EMA widths maximize the flow's log-likelihood on held-out assemblies.

	Pretraining (PT)	Finetuning (IC)
Data	100M 100-part assemblies	1M rendered assemblies
Part counts	prefixes of 2–100	2–100, balanced
Steps, batch	1M, 4,096	200k, 2,048
Muon LR (S/B/L)	$2.4/1.2/0.9 \times 10^{-3}$	$7.1/3.5/2.7 \times 10^{-4}$
AdamW LR	4×10^{-4}	1.4×10^{-4}
Flow time t	logit-normal, μ=1.2, σ=1.5 [Abramson et al., 2024]	logit-normal, μ=0, σ=1.5
Image encoder	–	DINOv3-L layer 18, 512×512
Resampler	–	Perceiver, 256 latents, 6 layers
Dropout	–	image 10%, materials 25%
Keypoint loss	–	β-NLL, weight 10

Both stages	
Optimizer	Muon [Jordan et al., 2024] for hidden matrices, AdamW otherwise
Schedule	inverse square root, 1k warmup
Weights	post-hoc EMA [Karras et al., 2024]
DiT-S/B/L	width 384/768/1024, depth 12/12/24, 33.3M/131.7M/460.4M parameters in PT

B Snapping

After sampling, we snap each sample. BrickNet detects its connections, and each detected connection is moved onto its exact seat if no part moves by more than 0.8 mm/2 LDU (rms) or 10°, no connection is lost, and no collision is added. It is otherwise left as-is. All reported results are after snapping.

Table 3: **Effect of snapping.** Snapped minus raw, paired by sample, with 95% CIs. Snapping raises validity and removes colliding pairs for every model while moving MM by at most 0.07 LDU, and it adds less to guided samples.

Model	Guid.	ΔValid	ΔColl.	ΔMM
ProcBrick				
PT DiT-S	–	1.8 [1.7, 1.9]	−4.99 [−5.07, −4.93]	0.02 [0.02, 0.02]
	DPS	0.8 [0.8, 0.9]	−5.10 [−5.18, −5.02]	0.01 [0.01, 0.01]
PT DiT-B	–	2.5 [2.3, 2.7]	−3.63 [−3.69, −3.58]	0.06 [0.06, 0.06]
	DPS	1.2 [1.1, 1.3]	−4.47 [−4.54, −4.40]	0.05 [0.05, 0.05]
PT DiT-L	–	3.0 [2.8, 3.2]	−2.80 [−2.85, −2.76]	0.07 [0.07, 0.07]
	DPS	1.9 [1.8, 2.0]	−5.66 [−5.74, −5.57]	0.06 [0.06, 0.06]
IC DiT-S	–	2.1 [2.0, 2.3]	−4.57 [−4.64, −4.50]	0.00 [0.00, 0.00]
	DPS	1.4 [1.3, 1.5]	−4.98 [−5.06, −4.90]	0.00 [0.00, 0.00]
IC DiT-B	–	3.0 [2.8, 3.1]	−3.48 [−3.54, −3.43]	0.02 [0.02, 0.02]
	DPS	1.9 [1.8, 2.0]	−4.36 [−4.43, −4.29]	0.02 [0.02, 0.02]
IC DiT-L	–	3.2 [3.1, 3.4]	−2.56 [−2.61, −2.51]	0.02 [0.02, 0.02]
	DPS	2.4 [2.3, 2.6]	−5.79 [−5.89, −5.70]	0.02 [0.02, 0.02]
BrickNet				
PT DiT-S	–	4.9 [4.1, 5.6]	−2.84 [−3.13, −2.58]	0.01 [0.01, 0.01]
	DPS	1.6 [1.3, 2.0]	−2.35 [−2.60, −2.12]	0.01 [0.01, 0.01]
PT DiT-B	–	6.9 [6.0, 7.8]	−2.27 [−2.51, −2.04]	0.04 [0.03, 0.04]
	DPS	2.5 [2.1, 2.9]	−2.13 [−2.37, −1.90]	0.03 [0.03, 0.04]
PT DiT-L	–	6.4 [5.5, 7.2]	−1.75 [−1.96, −1.55]	0.05 [0.04, 0.06]
	DPS	3.1 [2.7, 3.6]	−2.59 [−2.91, −2.28]	0.04 [0.04, 0.05]
IC DiT-S	–	7.5 [6.4, 8.7]	−2.73 [−2.98, −2.47]	−0.01 [−0.01, 0.00]
	DPS	4.3 [3.5, 5.1]	−2.36 [−2.61, −2.13]	0.00 [−0.01, 0.00]
IC DiT-B	–	9.3 [7.9, 10.6]	−2.21 [−2.45, −1.99]	0.00 [−0.01, 0.00]
	DPS	4.2 [3.5, 4.9]	−2.12 [−2.38, −1.88]	0.00 [0.00, 0.01]
IC DiT-L	–	7.0 [5.9, 8.0]	−1.70 [−1.91, −1.51]	−0.01 [−0.01, 0.00]
	DPS	4.4 [3.7, 5.0]	−2.65 [−2.98, −2.35]	0.00 [−0.01, 0.00]

C Connectivity Field Details

The seat distance of a candidate seat h between parts a and b is

$$c_h = \min_{\theta,\, Q} \frac{\mu_a\, d_a^2 + \mu_b\, d_b^2}{\mu_a + \mu_b}, \tag{10}$$

where d_a and d_b are the displacements (Eq. (3)) that bring the two parts onto the allowed relative transform with free coordinates θ, and Q is a common rigid motion of the pair, solved in closed form [Horn, 1987]. The taper w_h is the quintic smoothstep [Perlin, 2002] of the distance between the connector origins (minus, for axles, the largest admitted slide offset), equal to 1 below 12 LDU and 0 beyond 16 LDU. The seat penalty is $q(c) = \tau H((c-\tau)/\tau)$ and the hinge is $\phi(x) = \Omega_0 H(x/\Omega_0)$ with $\Omega_0 = (N-1)\rho$, where $H(x) = x^3/(x^2+1)$ for $x > 0$ and $H(x) = 0$ otherwise. The weak conductance between every pair of parts is $1/(16N)\,\mathrm{LDU}^{-2}$. We use $\rho = \tau = 10^{-8}\,\mathrm{LDU}^2$. On 1,000 held-out procedural assemblies, E and its gradient are exactly zero.

D Guidance Hyperparameters

We select settings on ProcBrick evaluation assemblies stratified by part count, with 8 samples each, varying one factor at a time with shared noise, and check them on BrickNet. Integrating rotations with the $\Delta t/(1-t)$ step of the training interpolant instead of the exponential schedule more than halves validity on this subset (PT DiT-L: 7.0% vs. 17.2% unguided).

Table 4: **Guidance hyperparameters.** Validity (%) of guided samples before snapping. Each change moves validity by at most 2.1 points. Top: changes from the initial setting. Bottom: changes from the final setting.

Setting	PT DiT-L	PT DiT-S
initial: γ=2, clip 1 LDU, t_{start}=0.7, p=4, taper 8–12 LDU	28.2	21.0
$\gamma = 1$ / 4	26.7 / 26.8	19.7 / 20.6
clip 0.5 / 2 LDU	28.5 / 27.8	21.2 / 21.4
$t_{\mathrm{start}} = 0.6$ / 0.8	27.5 / 28.2	20.9 / 21.5
final: taper 12–16 LDU	**28.6**	**21.5**
no clip	26.5	21.5
$t_{\mathrm{start}} = 0$	26.8	–

E Full Results

Table 5: **Full results.** Validity at collision allowances of 1.1 and 1.0 LDU, and MM, with 100 samples per structure. Validity at 1.0 LDU is nearly identical, and guidance changes MM by at most 0.3 LDU.

Model	Guid.	V@1.1	V@1.0	MM
ProcBrick				
PT DiT-S	–	5.5	5.5	72.45
	DPS	7.7	7.7	72.13
PT DiT-B	–	7.8	7.7	72.38
	DPS	10.9	10.8	72.21
PT DiT-L	–	10.3	10.3	72.56
	DPS	13.5	13.3	72.53
IC DiT-S	–	6.5	6.5	53.73
	DPS	8.9	8.8	53.64
IC DiT-B	–	10.1	10.1	48.38
	DPS	13.0	12.9	48.37
IC DiT-L	–	13.4	13.4	47.93
	DPS	15.8	15.6	47.96
BrickNet				
PT DiT-S	–	13.6	13.4	58.41
	DPS	18.0	17.9	58.12
PT DiT-B	–	19.8	19.7	58.65
	DPS	25.9	25.7	58.49
PT DiT-L	–	25.5	25.3	58.85
	DPS	31.7	31.5	58.78
IC DiT-S	–	19.4	19.2	34.90
	DPS	24.4	24.2	34.82
IC DiT-B	–	26.1	26.0	31.47
	DPS	31.7	31.5	31.43
IC DiT-L	–	30.5	30.3	30.95
	DPS	35.1	34.8	30.94

F Inference Time

Guidance makes one sample 1.5–3.4 times slower, almost entirely from work on the GPU rather than the energy; with batching on ProcBrick, it slows the IC models by at most 1.15 times. From 2–10 to 71–100 parts, time per sample rises by at most 1.4 times.

Table 6: **Inference time.** Seconds to sample one assembly on one H100 GPU.

	Seconds per sample		Samples per second	
Model	Unguided	DPS	Unguided	DPS
PT DiT-S	0.09	0.30	919	553
PT DiT-B	0.11	0.33	374	286
PT DiT-L	0.25	0.62	125	98
IC DiT-S	0.39	0.68	106	97
IC DiT-B	0.51	0.75	53	47
IC DiT-L	1.00	1.49	19	17

G Ablations

Table 7: **Ablations.** Variants of IC DiT-S, without guidance. Training without the pretrained model collapses validity, and less rendered data lowers it moderately. 10 samples per structure (100 for the reference rows).

Variant	Valid	Comp.	MM
ProcBrick, IC DiT-S			
IC DiT-S	6.5 [6.1, 6.9]	10.46	53.73
from scratch	1.2 [1.0, 1.4]	32.15	50.43
50% renders	6.3 [5.9, 6.7]	10.62	54.13
25% renders	5.8 [5.4, 6.3]	11.07	55.24
BrickNet, IC DiT-S			
IC DiT-S	19.4 [16.8, 22.1]	6.25	34.90
from scratch	2.3 [1.3, 3.5]	16.51	33.83
50% renders	18.1 [15.6, 20.7]	6.34	35.66
25% renders	16.2 [13.7, 18.8]	6.58	37.76

H ProcBrick Details

Generation. A placement is rejected if two parts interpenetrate by more than 1 LDU, unless they are joined by a fixed connection. Part-type frequencies are tilted toward larger parts and calibrated per connector family. The seed part follows the occurrence frequencies of part types in human-designed models. The acceptance test penalizes drops in packing density, and its targets for connector joins and connected neighbors per added part span roughly the median to the upper quantiles of human-designed models.

Splits. All 100M pretraining assemblies pass the BrickNet collision check on every prefix. From a separate pool, we draw disjoint sets of 1M image-training, 9,900 evaluation, and 1,000 geometry-validation assemblies.

Appearance and rendering. Each part's material is one of 260 renderer materials, drawn by a policy fitted to the palettes of 53,431 human-designed models. We render at 1024×1024 on a transparent background, lit by one of 964 environment maps or, for 10% of images, a studio setup. During finetuning, we composite renders onto ImageNet [Deng et al., 2009] or plain backgrounds and apply random camera roll, zoom, and offset, random erasing, photometric jitter, and sensor noise.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Secs. 1 and 4.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Sec. 5.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: No theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Secs. 3 and 4 and the appendix.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, checkpoints, and data will be released.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Sec. 4 and the appendix.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Bootstrap 95% confidence intervals; see Sec. 4.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: One or two B200 GPUs per model for pretraining and one to eight for finetuning; inference time in Sec. F.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conforms to the Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: No foreseeable negative societal impact.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No high-risk models or data.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: LDraw [Jessiman, 1995–2026] (CC BY), BrickNet [Kulits and Schmid, 2026], DINOv3 [Siméoni et al., 2025], and ImageNet [Deng et al., 2009] are cited and used under their licenses.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code, checkpoints, and ProcBrick will be released with documentation.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the method.