Table 1: **3D part completion.** Reconstruction quality of the parts and whole object. *reproduced.

Method	3D Mask	Parts			Whole Object		
		IoU ↑	F-Score↑	CD↓	IoU ↑	F-Score↑	CD↓
HoloPart* [58]	✓	0.658	0.836	0.065	0.821	0.945	0.018
PartGen [5]	✗	0.614	0.812	0.121	0.779	0.921	0.033
AutoPartGen	✗	0.665	0.861	0.047	0.832	0.967	0.012

masks-to-part scenario, y is the image I as well as the masked image $J^{(k)} = M^{(k)} \odot I$, denoting which parts should be generated next.

Knowledge of the previously generated parts $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(k-1)}$ is essential as this allows the model to ensure that the next part fits together well with the previously generated ones. Furthermore, in all cases we consider, the evidence y also provides some evidence on the overall shape of the object. As suggested in Fig. 3, we can represent the union of parts by simply concatenating their latent representations. However, for compactness, we found it useful to fuse their codes into one, which we obtain as: $\boldsymbol{z}^{(1,\ldots,k-1)} = E\left(\cup_{j=1}^{k-1} \text{sample}_N\, D(\cdot \mid \boldsymbol{z}^{(j)})\right)$, where sample_N is a function that samples N points from the surface of the object defined by the zero level set of the SDF function $D(\cdot|\boldsymbol{z}^{(k)})$.

We found it optional but useful to pin down the overall object by adding to the evidence y a code $\tilde{\boldsymbol{z}}$ for the object as a whole, which is either given outright (object-to-part, $\tilde{\boldsymbol{z}} = E(\text{sample}_N\, \hat{\boldsymbol{x}})$) or can be obtained by the model itself (image-to-part and masks-to-part, $\tilde{\boldsymbol{z}} \sim p(\boldsymbol{z} \mid I)$).

With all this, we learn a conditional generator model

$$\boldsymbol{z}^{(k)} \sim p(\boldsymbol{z}^{(k)} | \tilde{\boldsymbol{z}}, \boldsymbol{z}^{(1,\ldots,k-1)}, y), \tag{1}$$

where $y = \phi$ for the object-to-part scenario, $y = I$ for the image-to-part scenario, and $y = (I, M^{(k)})$ for the masks-to-part scenario. The generation process stops when all the input masks have been processed, if available, or when the model outputs a special `[EoT]` token, representing empty shape.

Based on Section 3.2, learning the distribution Eq. (1) amounts to learning a velocity field $\hat{\boldsymbol{v}}(t, \boldsymbol{z}_t \mid \tilde{\boldsymbol{z}}, \boldsymbol{z}^{(1,\ldots,k-1)}, y)$. During inference, we use *classifier-free guidance* (CFG) [18] to modulate the strength of the conditioning. In the most general case, the model is conditioned by the overall (partial) object $\tilde{\boldsymbol{z}}$, the previously generated parts $\boldsymbol{z}^{(1,\ldots,k-1)}$, and a masked image pair $y = (I, J^{(k)})$. We modulate the importance of the geometric and visual conditioning as follows:

$$\begin{aligned}\hat{\boldsymbol{v}}_{\text{CFG}}(t, \boldsymbol{z}_t \mid \tilde{\boldsymbol{z}}, \boldsymbol{z}^{(1,\ldots,k-1)}, y) = w_{\text{img}} \left(\hat{\boldsymbol{v}}(t, \boldsymbol{z}_t \mid \tilde{\boldsymbol{z}}, \boldsymbol{z}^{(1,\ldots,k-1)}, I, J^{(k)}) - \hat{\boldsymbol{v}}(t, \boldsymbol{z}_t \mid \tilde{\boldsymbol{z}}, \boldsymbol{z}^{(1,\ldots,k-1)})\right) \\ + w_{\text{geom}} \left(\hat{\boldsymbol{v}}(t, \boldsymbol{z}_t \mid \tilde{\boldsymbol{z}}, \boldsymbol{z}^{(1,\ldots,k-1)}) - \hat{\boldsymbol{v}}(t, \boldsymbol{z}_t, \emptyset)\right) + \hat{\boldsymbol{v}}(t, \boldsymbol{z}_t, \emptyset)\end{aligned} \tag{2}$$

where w_{img} and w_{geom} modulate, respectively, image and geometry conditioning. The different inputs are implemented by appending tokens, which are then cross-attended by a transformer neural network computing the flow velocity. Hence, to suppress an input we simply replace it with dummy tokens. In the same way, we randomly drop some input at training time to allow the model to learn to use any required subset of the inputs.

Discussion Here, we contrast our model to prior works and justify its design. The most straightforward approach to part generation is to sample each part $\boldsymbol{x}^{(k)}$ independently from a 'marginal' distribution $p(\boldsymbol{x}^{(k)})$. However, this model lacks a mechanism to tie the parts together and would result in a soup of random, uncoordinated parts. The simplest such mechanism is to provide evidence y for the overall shape of the 3D object. For instance, in the image-to-3D case, $y = I$ could be a 2D image of the object, and we may sample parts from the conditional distribution $p(\boldsymbol{x}^{(k)} \mid I)$. While I constrains the shape and position of the possible parts, these are still quite ambiguous. This explains why PartGen [5] conditions part generation on a multi-view image $y = I_{MV}$ of the 3D object $\boldsymbol{x}$, and HoloPart [58] starts from a (partial) 3D reconstruction $y = \hat{\boldsymbol{x}}$ of the object itself.

Even then, the reconstruction context y is likely insufficient because there is no indication of *which* part should be reconstructed next. We could sample the parts in a random order, but this would not be very efficient. Furthermore, because the part decomposition is not unique, we would still need to extract a coherent subset of parts from the 'part soup' so obtained.