

Farm3D: Learning Articulated 3D Animals by Distilling 2D Diffusion

Tomas Jakab* Ruining Li* Shangzhe Wu Christian Rupprecht Andrea Vedaldi

Visual Geometry Group, University of Oxford

{tomj, ruining, szwu, chrisr, vedaldi}@robots.ox.ac.uk

farm3d.github.io

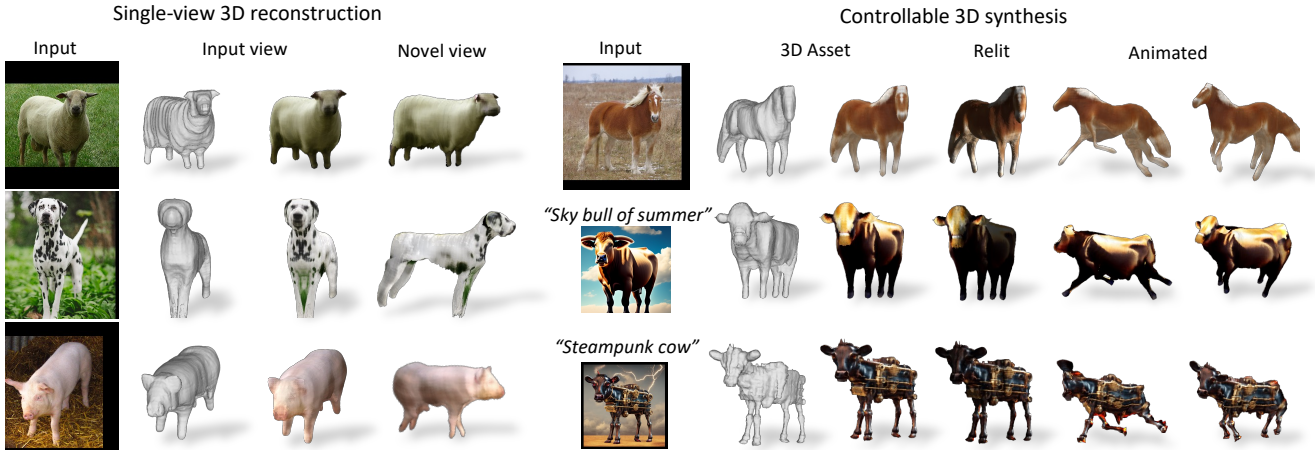


Figure 1: **Learning to Reconstruct 3D Animal Categories by Virtual Training from 2D Diffusion.** Our method learns to reconstruct articulated and textured animals from single images, real or generated, using only virtual supervision from an off-the-shelf diffusion-based 2D image generator. We show results on cows, horses, sheep, pigs, and dogs, which demonstrate that the method generalises to a wide range of animal categories without any modifications. Moreover, we demonstrate that our method can be used for controllable 3D synthesis: We can relight our generated 3D assets, swap their textures by conditioning on another input, and animate our articulated shapes, giving us greater control over the generated assets.

Abstract

We present Farm3D, a method to learn category-specific 3D reconstructors for articulated objects entirely from “free” virtual supervision from a pre-trained 2D diffusion-based image generator. Recent approaches can learn, given a collection of single-view images of an object category, a monocular network to predict the 3D shape, albedo, illumination and viewpoint of any object occurrence. We propose a framework using an image generator like Stable Diffusion to generate virtual training data for learning such a reconstruction network from scratch. Furthermore, we include the diffusion model as a score to further improve learning. The idea is to randomise some aspects of the reconstruction, such as viewpoint and illumination, generating synthetic views of the reconstructed 3D object, and have the 2D network assess the quality of the resulting image, providing

feedback to the reconstructor. Different from work based on distillation which produces a single 3D asset for each textual prompt in hours, our approach produces a monocular reconstruction network that can output a controllable 3D asset from a given image, real or generated, in only seconds. Our network can be used for analysis, including monocular reconstruction, or for synthesis, generating articulated assets for real-time applications such as video games.

1. Introduction

The explosive success of generative AI has led to exciting progress in image generation, with methods such as DALL-E [21], Imagen [24] and Stable Diffusion [22] producing high-quality images from textual prompts. This success is likely to transfer beyond 2D data. DreamFusion [20] has recently shown that one can distil high-quality 3D models from a text-to-image generator. While the generator

*Equal contribution.

is not trained with 3D capabilities, it nevertheless contains sufficient information to recover a 3D shape.

In this paper, we show that one can extract more from a text-to-image generator and obtain from it *articulated models of 3D object categories*. That is, our goal is not to extract a single 3D asset (DreamFusion), but rather a statistical model of an entire category of articulated 3D objects (*e.g.*, cows, sheep, horses) capable of reconstructing an animatable 3D asset, easily applicable to AR/VR, gaming and content creation, from a single image, real or synthesised.

We approach this problem as learning a network that, given a single image of an object, predicts a corresponding articulated 3D model. Prior works rely on real data to learn such reconstruction networks [14, 42, 35], but here we suggest using instead virtual data generated from a 2D diffusion model such as Stable Diffusion. This has several advantages. First, the 2D image generator tends to produce realistic and clean samples of the object category, implicitly curating the training data and simplifying learning. Second, the 2D generator implicitly provides *virtual views* of any given object instance via distillation [20], which further disambiguates learning. Third, it makes the approach more flexible by avoiding the need to collect (and potentially curate) real data.

Our approach, which we call Farm3D, is complementary to 3D generators such as DreamFusion [20], RealFusion [18] and Make-a-video-3D [27], which generate a single 3D asset, static or dynamic, via test-time optimisation, starting from text or an image, and requiring hours. At test time, our network performs reconstruction from a single image in a feed-forward manner in seconds and produces not a fixed 3D or 4D artefact, but an articulated 3D model that can be controlled (*e.g.* animated, relighted). Our approach is not only useful for synthesis but for *analysis* too, because, despite being trained exclusively on virtual data, the reconstruction network does generalise to real images. For example, we can envisage applications to animal behaviour research and conservation.

Farm3D rests on two key technical **contributions**. First, we show that, via prompt engineering, Stable Diffusion can be induced to generate a large training set of relatively clean images of an object category for the purpose of learning articulated 3D models. We show that these images can be used to bootstrap MagicPony [35], a state-of-the-art method for monocular reconstruction of articulated objects.

Second, we show that the Score Distillation Sampling (SDS) loss of [20] can be extended to obtain synthetic multi-view supervision to train a photo-geometric autoencoder, in our case MagicPony, instead of fitting a single radiance field model. As the photo-geometric autoencoder decomposes the object into different factors of the image formation (*i.e.*, articulated shape, appearance, camera viewpoint, and illumination), it allows us to resample some of

these factors (*e.g.*, viewpoint and illumination) to generate new synthetic views of the same object. These synthetic views are then fed into the SDS loss to receive a gradient update, which is back-propagated to the learnable parameters of the autoencoder.

We assess Farm3D qualitatively in terms of its 3D reconstruction and generation capabilities. Furthermore, because Farm3D is capable of reconstruction in addition to generation, we can also test it quantitatively on analysis tasks such as semantic keypoint transfer. We demonstrate comparable or even superior performance to various baselines despite the fact that the model does not use any real image for training and thus avoids time-consuming data collection and curation.

2. Related Work

Weakly-supervised 3D Object Learning. While reconstructing deformable 3D objects traditionally requires simultaneous multi-view captures [6], several recent works have demonstrated that it is possible to learn 3D models of deformable objects purely from single-view image collections, with some form of geometric supervision in addition to segmentation masks, such as keypoint annotations [10, 9], category-specific template shapes [5, 13, 12], semantic correspondences distilled from image features [14, 42, 35] and/or strong assumptions like symmetry [37, 36]. Alternatively, with a known prior viewpoint distribution, one can also use a generative adversarial framework to learn simpler 3D objects, such as faces and cars [19, 25, 1, 2]. Researchers have also leveraged monocular videos as training data, with additional temporal signals for learning [38, 39, 34, 40, 41]. Although impressive results have been shown, many of them still rely on heavily curated category-specific data for training, restricting the model to a only small number of categories. Here, we introduce a method for distilling 3D objects from large 2D diffusion models, which can potentially generalise to a wide range of object categories.

Diffusion Models. Recent years have seen a flurry of diffusion models [28, 29, 7, 30], which have become the key-stone of the new generation of text-to-X generative models, where X can be images [21, 24, 22], videos [26], vector graphs [8], audio [17] and so on. In particular, these models can generate complex high-fidelity samples by learning to reverse a diffusion process, *i.e.*, gradually removing synthetically added noise until the image is recovered. Text-to-image diffusion models [21, 24, 22], in particular, introduce textual conditioning to these generative models, which offers a powerful interface for general controllable image generation. Although these models have demonstrated some level of compositionality and controllability, it is unclear what kind of 3D information is encoded in these learned

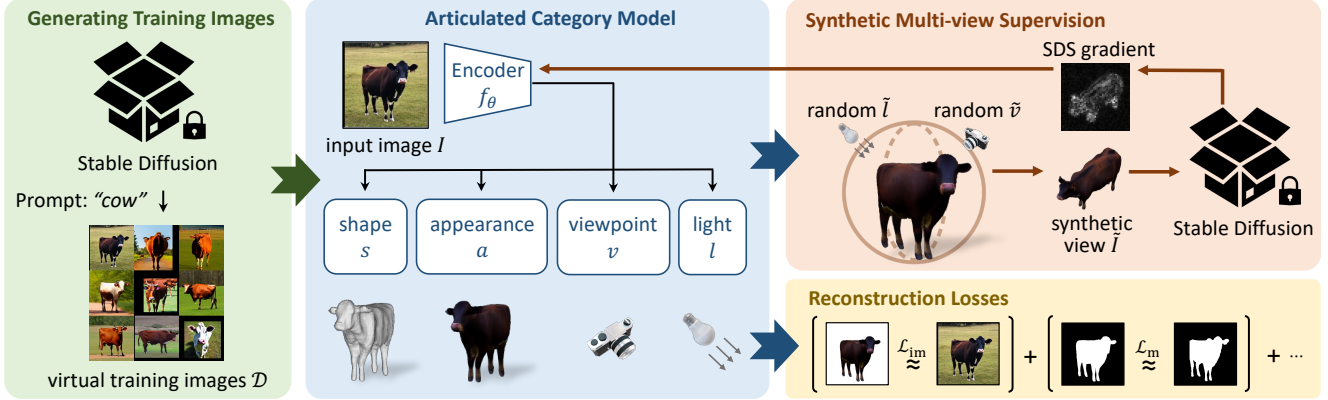


Figure 2: **Training Pipeline.** We prompt Stable Diffusion for virtual views of an object category that are then used to train a monocular articulated object reconstruction model that factorises the input image of an object instance into articulated shape, appearance (albedo and diffuse and ambient intensities), viewpoint, and light direction. During training, we also sample synthetic instance views that are then “critiqued” by Stable Diffusion to guide the learning.

image synthesis models.

Distilling 3D Models from Image Diffusion Models.

Several recent works started exploring the extraction of 3D information from large pre-trained 2D image diffusion models. In particular, DreamFusion [20] and Magic3D [15] have demonstrated the possibility of generating diverse full 3D models from text queries by prompting pre-trained image diffusion models. Make-A-Video3D [27] adopts a similar strategy for generating 4D dynamic scenes, and RealFusion [18] instead extends the pipeline to reconstructing 3D objects in real photos.

Our method differs from these approaches by learning an articulated category-level model. This has several advantages. First, it predicts the 3D shape in a single forward pass, eliminating the need for lengthy optimisation as required by other methods. Second, the category-level model enables us to directly relate semantically corresponding points on the surface of objects within the same category, which enables numerous applications, such as texture swapping (conditioning the texture on another input image) and image understanding (as demonstrated in Table 1). Third, our method learns articulated shapes, providing greater control over the shape generation that can be used for animation.

3. Method

Our goal is to learn an articulated 3D model of an object category, such as animals (cows, sheep, horses, *etc.*), using exclusively virtual training data generated by a pre-trained 2D image generator. We base our model on the recent MagicPony framework [35], which also serves as a baseline in experiments (Sec. 3.1). MagicPony is designed to learn articulated 3D objects from real image collections; here, we show how this can be extended to use virtual train-

ing data, replacing real data. We do so by generating virtual image samples via prompting (Sec. 3.2) and by modifying the MagicPony training objective to integrate the 2D diffusion model as a critic (Sec. 3.3). We provide a schematic overview of the method in Fig. 2.

3.1. Articulated Category Model

MagicPony [35] learns a monocular reconstruction network through a 3D-aware autoencoding framework, which we summarise for completeness. On a high level, the model f_θ receives as input a single RGB image $I \in \mathbb{R}^{3 \times H \times W}$ and outputs a set of photo-geometric parameters $(s, a, v, l) = f_\theta(I)$ describing the object contained in the image, where θ are the model parameters. Here, s is the object shape (accounting for a category-level prior shape, an instance-specific deformation, and image-specific bone articulations), a is the appearance (accounting for albedo and diffuse and ambient intensities), $v \in SE(3)$ is the object viewpoint (expressed as a rotation and a translation with respect to the camera), and $l \in \mathbb{S}^2$ is the prominent direction of the illuminant.

The photo-geometric encoder f_θ is paired with a rendering function $\hat{I} = R(s, a, v, l)$ which reproduces the image of the object. Key to the method is the fact that R is a *hand-crafted* (not learned) differentiable renderer, which implicitly assigns s, a, v, l their photo-geometric meaning.

MagicPony learns from a collection of monocular images and individual frames from videos but assumes a curation process that crops the images around the objects of interest and excludes occluded and truncated instances. It also assumes a segmentor to obtain the object masks $M \in \{0, 1\}^{H \times W}$, such as PointRend [11]. Given the resulting dataset \mathcal{D} of training pairs (I, M) , MagicPony min-

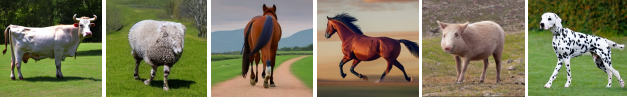


Figure 3: **Virtual Training Images.** Examples of virtual training images generated with Stable Diffusion. The generated animals are typically without occlusions but sometimes anatomically incorrect (e.g., columns 2 and 3), but our model is robust to this and learns plausible 3D shapes.

imises the objective:

$$\mathcal{L}(\theta|\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(I,M) \in \mathcal{D}} \mathcal{L}(f_\theta(I)|I, M) + \mathcal{R}(f_\theta(I)),$$

where $\mathcal{L}(f_\theta(I)|I, M)$ checks how well the predicted object model reconstructs the input image I , the input mask M , and the ViT-DINO features $\Phi(I)$, and the regulariser $\mathcal{R}(f_\theta(I))$ regularises the prior shape (Eikonal loss for an SDF), and shrinks the amount of instance-specific deformation and articulation.

3.2. Generating Training Images via Prompting

While in prior category-specific work, the training data \mathcal{D} consists of lightly-curated real images, here we propose to replace them with entirely *virtual* images of a target object category obtained from an off-the-shelf image generator. We do so by prompting the Stable Diffusion model [22], a high-quality 2D diffusion model, with text specific to the object category. All training data are generated using the exactly same prompts, with the only variation being the category name.

When simply prompted for images of an object category, e.g. ‘cow’, Stable Diffusion generates mostly frontal and side views of the object. We hypothesise that this is due to the bias contained in its training data, an observation similar to that of [20]. We found that naive view-dependent prompting [20] for $\{\textit{side}, \textit{front}, \textit{back}\}$ does not work well with Stable Diffusion (as opposed to Imagen [24] used in [20]). Instead, we found that prompting Stable Diffusion for an animal category “walking away from the camera” biases the generation well enough to obtain images that have diverse view coverage. More details on the textual prompt design are included in the supplementary material.

Formally, Stable Diffusion uses an autoencoder h that maps an image I to a latent code $z_0 = h(I) \in \mathbb{R}^{D' \times H_h \times W_h}$, and learns a conditional distribution $p(z_0|y)$ of the latent code conditioned on a textual prompt y . For a typical diffusion model, it does so by considering the sequence of noised signals $z_t = \alpha_t z_0 + \sigma_t \epsilon_t = \alpha_t h(I) + \sigma_t \epsilon_t$ where ϵ_t is a normally distributed noise, $\alpha_t = \sqrt{1 - \sigma_t^2}$ and $\sigma_t \in (0, 1)$, $t = 0, 1, \dots, T$ is an increasing sequence of

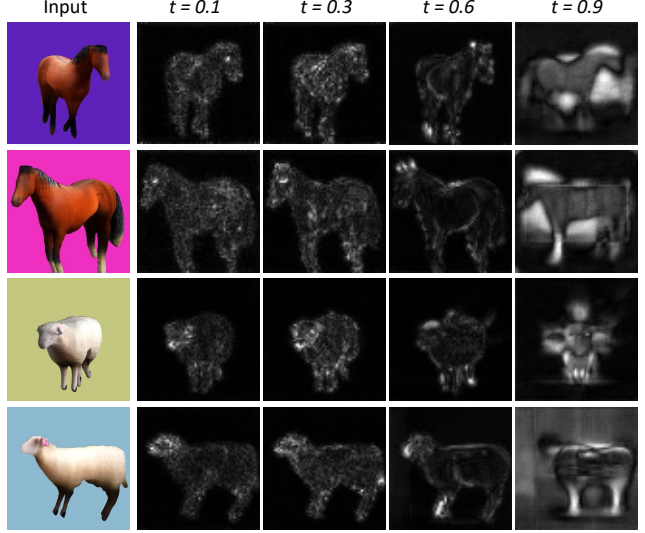


Figure 4: **Noise Scheduling and SDS Gradient.** We show four rendered images \hat{I} , obtained by first sampling a random viewpoint and illumination as for training our model. Then, we pick a fixed noise sample ϵ and show the SDS gradient $(\hat{\epsilon}_t(z_t|y) - \epsilon)(\partial h / \partial \hat{I})$ used to update \hat{I} in Eq. (2) for different values of σ_t . Because ϵ is fixed, $z_t = \alpha_t h(\hat{I}) + \sigma_t \epsilon$ only depends on σ_t . Large noise levels ($t = 0.9$) generate an update which is essentially independent of the input image \hat{I} . Lower noise levels provide more meaningful gradients and lead to more stable training as demonstrated in Fig. 9.

noise standard deviations from $\sigma_0 \approx 0$ to $\sigma_T \approx 1$. The Stable Diffusion model is then a ‘denoising network’ $\hat{\epsilon}_t(z_t|y)$ that approximates the noise content ϵ_t of z_t , trained to minimise a loss of the type:

$$\mathcal{L}(\hat{\epsilon}) = \mathbb{E}_{t, \epsilon, I, y} [\|\hat{\epsilon}_t(\alpha_t h(I) + \sigma_t \epsilon_t | y) - \epsilon_t\|^2], \quad (1)$$

averaged over noise level t , noise samples ϵ and empirical samples (I, y) (captioned images). Finally, in order to draw an image sample I from the learned distribution, one draws a random sample z_T from a normal distribution and then progressively denoise it with $z_{t-1} = \alpha_t^{-1}(z_t - \sigma_t \hat{\epsilon}_t(z_t))$ to obtain z_0 and eventually $I = h^{-1}(z_0)$.

In our case, $\hat{\epsilon}$ is obtained from an off-the-shelf pre-trained model. We use it to draw image samples $I \sim p(I|y)$ where y are prompts constructed as described above.

3.3. Distilling a 3D Reconstructor

In addition to training data generation from scratch, we can also use the 2D image generator as a form of *synthetic multi-view supervision* for training the reconstruction network f_θ . In order to do so, given an example image I , we first obtain an estimate $(s, a, v, l) = f_\theta(I)$ of the object’s photo-geometric parameters. Then, we randomly sample a *new camera viewpoint* \tilde{v} and a *new light direction* \tilde{l} , and



Figure 5: **Single-View Real Image Reconstruction.** We present the reconstructed mesh along with predicted texture from the input view and two other views. Our method reconstructs the shape of a wide range of categories to their fine details such as legs and ears despite not being trained on any real images.

render the corresponding image:

$$\tilde{I} = R(s, a, \tilde{v}, \tilde{l}) = R \circ \text{subst}_{\tilde{v}, \tilde{l}} \circ f_{\theta}(I).$$

Here the operator $\text{subst}_{(\tilde{v}, \tilde{l})}$ replaces the predicted viewpoint v and light l from the output of $f_{\theta}(I)$ with the new values \tilde{v} and \tilde{l} . We have no direct supervision for the resulting image \tilde{I} , but, similarly to [20], we can use the 2D image generator as a critic to judge whether \tilde{I} ‘looks correct’.

Concretely, this is achieved through a new variant of the *Score Distillation Sampling* (SDS) loss [20]. Given a training image I , the corresponding caption y , and randomly-sampled viewpoint \tilde{v} and lighting \tilde{l} parameters, the *gradient* of the distillation loss is:

$$\nabla_{\theta} \mathcal{L}_{\text{SDS}}(\theta | \tilde{v}, \tilde{l}, I, y) = \mathbb{E}_{t, \epsilon} \left[w_t \cdot (\hat{\epsilon}_t(z_t | y) - \epsilon_t) \frac{\partial z_t}{\partial \theta} \right], \quad (2)$$

where $w_t > 0$ is a weight factor, ϵ_t a noise sample, and

$$z_t = \alpha_t \cdot \left(h \circ R \circ \text{subst}_{\tilde{v}, \tilde{l}} \circ f_{\theta} \right) (I) + \sigma_t \epsilon_t$$

is the noised version of the object image obtained from the new viewpoint and lighting. Based on this definition, the

derivative in Eq. (2) is, with a slight abuse of notation,

$$\frac{\partial z_t}{\partial \theta} = \alpha_t \cdot h'(\tilde{I}) \cdot \frac{\partial R}{\partial (s, a)} \Big|_{(s, a, \tilde{v}, \tilde{l})} \cdot \frac{\partial [f_{\theta}]_{sa}}{\partial \theta} \Big|_I.$$

The first factor α_t comes from the diffusion scaling. The second factor is the derivative of the Stable Diffusion latent code z_0 with respect to the coded image I . The third factor is the derivative of the rendering function with respect to the shape and appearance components only. The last factor is the derivative of the shape s and appearance a predictions by the model f_{θ} . Note that the viewpoint \tilde{v} and the lighting \tilde{l} are not included in the derivative because they are not estimated by f_{θ} but sampled.

Noise Scheduling. We found that scheduling the noise properly in Eq. (2) is critical. If the noise is too large, the noised latent image bears little to no relation to the one which is actually reconstructed; as a result, the feedback from the SDS loss points in a novel direction, reconstructing ‘any’ object compatible with the prompt y (as in DreamFusion [20]), instead of improving the reconstruction of the *specific* object contained in the input image I . As motivated in Fig. 4, in practice, we sample $t \sim \mathcal{U}(0.02, t_{\max})$ where $t_{\max} = 0.6$ in the definition of Eq. (2), as opposed

to DreamFusion that uses $t_{max} = 0.98$. The choice of t_{max} is further supported by an ablation study in Fig. 9.

Further Details. Computing SDS gradient (2) has a relatively large memory footprint and requires a halved batch size, which deteriorates training stability. We found it helpful in practice to apply SDS guidance every second iteration. Specifically, at every second iteration, camera positions \tilde{v} are randomly sampled in spherical coordinates, with an elevation angle $\phi_{cam} \in [-10^\circ, 90^\circ]$, an azimuth angle $\theta_{cam} \in [0^\circ, 360^\circ]$, and a distance from the origin in [9, 11]. For reference, we initialise the prior shape as an ellipsoid with axis lengths 2.1, 2.1 and 1.05. We found that randomly sampling the camera field of view (as in DreamFusion) did not improve results. Instead of sampling point lights (as in DreamFusion), we assume a distant directional light following MagicPony, and sample a random light direction $\tilde{l} \sim \mathcal{N}(\tilde{v}/\|\tilde{v}\|, \mathbf{I})$, where $\tilde{v} \in \mathbb{R}^3$ is the sampled camera viewpoint. This sampling strategy, first used in DreamFusion, ensures that the side of the object facing towards the camera is mostly illuminated.

We render 256×256 images, and like in DreamFusion, randomly alternate between a shaded texture, texture-less shading, and albedo-only texture (*i.e.*, no lighting). We also use the same classifier-free guidance (with a classifier guidance strength of 100) as it improves the regularity of the learned shape.

3.4. Discussion

Our model is related to DreamFusion [20], and in fact, borrows some ideas from it, but with several fundamental differences. Both methods start from a text-conditional image generator network ϵ . DreamFusion distils from the network ϵ a *single* 3D model V (a neural radiance field), corresponding to a textual prompt y . We learn instead a *monocular reconstruction network* f_θ , or in other words, a category-level prior 3D model. This network is, at test time, capable of inferring a 3D model $(s, a, v, l) = f_\theta(I)$ from a single image I of a new instance in a feed-forward manner, without further optimisation, in seconds. This should be contrasted to inference in DreamFusion, which requires hours. Our method can also be used to reconstruct photographs of real objects, which DreamFusion is not capable of, and the output is not a radiance field, but an articulated model of the object. Furthermore, since our output is a *mesh*, it can be easily used in applications, which is harder to do with a radiance field.

The main shortcoming of our approach compared to test-time distillation is the limited generality, which comes at the cost of developing a category-specific model. Specifically, we make assumptions about the object categories we would like to learn, including the topology of the articulated skeleton (*e.g.*, 4 legs), and focus on reconstructing single objects, rather than a full 3D scene.

4. Experiments

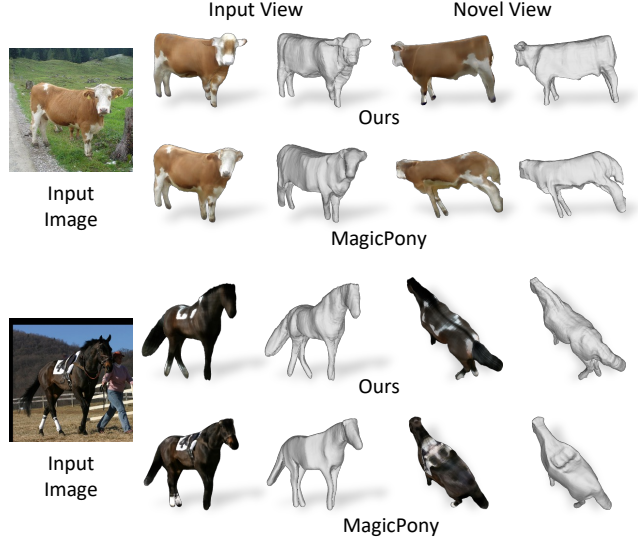


Figure 6: **Comparison with MagicPony on Real Cow and Horse Images.** Our model predicts more plausible 3D shapes than MagicPony [35] without being trained on any real images.

Table 1: **Keypoint Transfer on PASCAL VOC.** We report PCK@0.1 (\uparrow better). Our method achieves comparable results to MagicPony and outperforms other approaches, without training on any real images. [†]The performance of weakly-supervised methods, such as MagicPony, significantly depends on the cleanliness of the training data as the performance drops when trained on an automatically pre-processed ImageNet subset.

Method	Horses	Cows	Sheep
UMR [14]	0.244	—	—
A-CSM [13]	0.329	0.263	0.286
MagicPony [35]	0.429	0.425	0.262 [†]
Ours (full model)	0.425	0.402	0.328
Ours w/o SDS	0.414	0.370	0.376

We conduct experiments on both single image 3D reconstruction (Sec. 4.3) and 3D asset generation (Sec. 4.4) on several animal categories, including cows, sheep, horses, pigs, and Dalmatian dogs, showcasing the generalisation capability of our method across a variety of articulated animal species. We also compare our approach both qualitatively and quantitatively to prior work, demonstrating that by training solely on virtual images, we can achieve comparable, if not superior, results.

4.1. Additional Technical Details

We follow the implementation of MagicPony [35] for the underlying articulated 3D representation. In particular, MagicPony represents the appearance of an object using a single feature vector and is unable to model high-fidelity details in the input image. Following [35], for the visualisations with texture rendering, we fine-tune the appearance network for 100 iterations on each test image to obtain sharper textures. This step is optional but results in a more faithful texture at a modest additional cost (the 3D shape prediction takes 0.5 seconds and the optional texture fine-tuning requires up to 30 seconds, vs. several hours of distillation approaches like DreamFusion). Notably, these textures still generalise to the unobserved parts of the object owing to the underlying symmetric neural field representation. All the technical details are included in the appendix.

4.2. Datasets

We generate 30k virtual views per object category using the procedure described in Sec. 3.2 to train our method without any real images. The segmentation masks are obtained with PointRend [11] and the samples containing likely occluded instances are automatically removed using simple heuristics, as detailed in the appendix. Fig. 3 shows a few examples of the pre-processed virtual training data.

For cows, sheep and horses, we evaluate our model qualitatively using images in PASCAL [4] and COCO [16] and quantitatively on the keypoint transfer task using annotations from PASCAL. For pigs and Dalmatian dogs, we show qualitative results using ImageNet [3] images and manually collected Internet images respectively. Since MagicPony [35] does not provide a model for sheep, we further collect 8k real training images from ImageNet sheep synsets and fine-tune their pre-trained horse model.

For all images, either obtained from public datasets or collected from the Internet, we follow the same pre-processing procedure as for our virtual data to obtain segmentation masks and remove truncated objects.

4.3. Single Image 3D Reconstruction

We evaluate the effectiveness of our method on the monocular 3D reconstruction task. The results, as shown in Fig. 5, demonstrate that our method is able to generalise across several animal categories, producing high-quality 3D meshes with faithful texture and accurate articulation, including reconstruction of fine-grained shapes such as legs and ears. We also provide additional results in the appendix to further support the effectiveness of our approach.

Qualitative Comparisons. We compare our model against the MagicPony [35] baseline, which is trained on a large collection of real images, including video frames, whereas our model learns purely from virtual training data

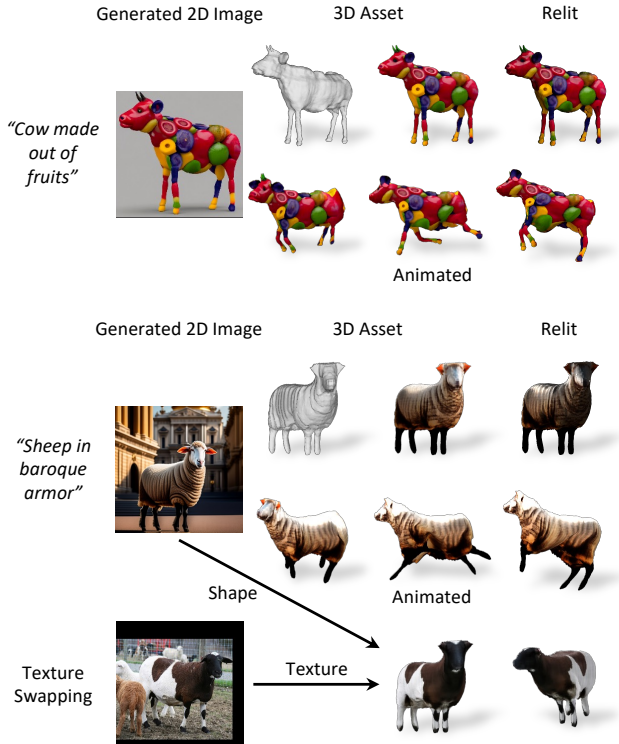


Figure 7: **Controllable 3D Synthesis.** Our method enables the generation of controllable 3D assets from either a real image or an image synthesised using Stable Diffusion. Once generated, we have the ability to adjust lighting, swap textures between models of the same category, and even animate the shape. For shapes that are out of the training distribution, in this case “cow made out of fruits” (top row), we use the optional test time shape adaptation as described in Sec. 4.4.

generated by Stable Diffusion. Fig. 6 shows a qualitative comparison of single image 3D reconstruction results. Despite not seeing any real images at training, our model reconstructs high-quality shapes, less overfitting to the input view. The articulated shape predictions are more plausible, particularly when viewed from novel views. This can be attributed to the stronger regularisation obtained by the distillation loss that virtually provides multi-view supervision during training. Our observations are further supported by the ablation study presented in Fig. 8.

Quantitative Comparisons. To quantitatively evaluate our approach, we employ the keypoint transfer task, which is a common metric used in weakly-supervised object reconstruction [14, 13, 35], on PASCAL [4]. In addition to MagicPony [35], we also compare with two other weakly-supervised 3D object learning methods, UMR [14] and A-CSM [13]. UMR does not model complex articulations and produces less accurate shapes in general. A-CSM, on the

other hand, reconstructs articulated poses, but is heavily limited by the input template shape, and often results in inaccurate poses. Our method outperforms both methods significantly, and achieves comparable or even superior results to MagicPony, despite not being trained on any real images.

MagicPony’s performance gap between sheep and other categories likely comes from the fact that the original models were trained on cleaner data (horses and cows) that have undergone aggressive manual curation, as described by the authors. On the other hand, to fine-tune MagicPony on sheep (from its pre-trained horse checkpoint), we simply pre-processed the sheep images from ImageNet with the same automatic pre-processing procedure as used for our virtual dataset, resulting in larger instance variation and more occlusion. This also highlights the sensitivity of weakly-supervised methods to the level of curation of the training data, underscoring the advantages of our approach. By using Stable Diffusion generation, our method obtains much cleaner training data, as demonstrated in Fig. 3.

Additionally, it is worth noting that while our approach that learns a category model can be used for image understanding, as we have demonstrated on the keypoint transfer, other distillation-based methods such as DreamFusion [20] or RealFusion [18] cannot be used for this purpose. RealFusion does provide a 3D reconstruction of real images, but cannot learn and exploit category-specific priors, and it still requires expensive inference time optimisation.

4.4. Controllable 3D Shape Synthesis

In addition to controlling articulation via the underlying skeleton model, our approach enables controllable 3D synthesis, as demonstrated in Fig. 7. The model can be conditioned on either an image or text. For text conditioning, such as “sheep in baroque armor”, Stable Diffusion is first used to generate an example image of the input prompt. The image is then fed into the model as done with real images.

Unlike distillation approaches that require hours of test-time optimisation, our approach produces the 3D shape in a single pass, enabling faster iterations when producing 3D assets. Notably, we can explicitly control articulation, lighting, and albedo (*i.e.*, texture swapping) using our method.

(Optional) Test Time Shape Adaptation. During test time, we can optionally fine-tune our category model on a single image. This allows us to adapt the model to shapes that are out of the training distribution, such as “cow made out of fruits” as demonstrated in Fig. 7. As this process is very fast (less than 30 seconds) this allows users to control the outputs of generative models in 3D, which is an exciting capability with many potential applications.

4.5. Ablations and Failure Cases

We conduct ablations to evaluate the effect of the SDS guidance on the quality of the learned pose and shape, both

quantitatively in Table 1 and qualitatively in Fig. 8. Our qualitative results illustrate that using SDS helps the method learn more plausible shapes and articulation.

We also perform an ablation study on the noise scheduling in Fig. 9, demonstrating that it is critical to training stability. Without our noise scheduling, the learning is unstable, and the model often collapses multiple times during training. We show typical failure cases in the appendix.

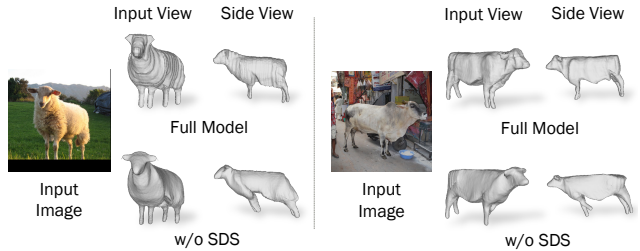


Figure 8: **Ablation Study on the SDS Loss.** The SDS loss uses synthetic views of the object instance which helps to achieve more consistent pose and articulation of the object.

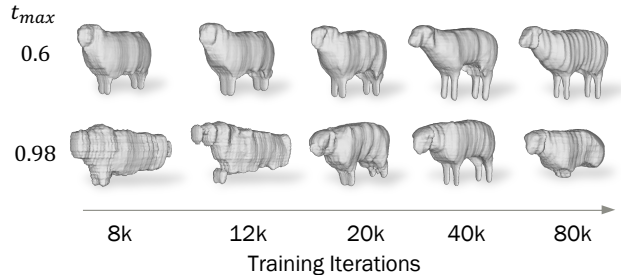


Figure 9: **Effect of our SDS Noise Scheduling on the Category-Specific Prior Shape as Training Progresses.** Sampling with $t_{max} = 0.6$ helps the SDS guidance stay more faithful to the input image, and hence stabilises training as can be seen from the learned prior shape.

5. Conclusions

We have presented Farm3D, a method for learning a monocular 3D reconstructor of an object category entirely from virtual data obtained from a diffusion-based off-the-shelf text-to-image generator. There are many enticing aspects of this approach: It avoids or reduces the need for collecting and curating real data to learn 3D objects, it learns 3D reconstruction networks that produce outputs in seconds at test time, and these networks can be used for monocular reconstruction (analysis) or for 3D asset generation (synthesis), producing articulated mesh models that can be easily used in applications.

Ethics. We use the PASCAL, ImageNet and MSCOCO datasets in a manner compatible with their terms. Some of these images may accidentally contain faces or other personal information, but we do not make use of these images or image regions. For further details on ethics, data protection, and copyright please see <https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html>.

Acknowledgements. We thank Laurynas Karazija, Sagar Vaze, and Luke Melas-Kyriazi for insightful discussions. Andrea Vedaldi and Christian Rupprecht are supported by ERC-CoG UNION 101001212. Tomas Jakab and Christian Rupprecht are (also) supported by VisualAI EP/T028572/1.

References

- [1] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 2
- [2] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2
- [3] 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. 7
- [4] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 7
- [5] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoints without keypoints. In *ECCV*, 2020. 2
- [6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2
- [8] Ajay Jain, Amber Xie, and Pieter Abbeel. VectorFusion: Text-to-svg by abstracting pixel-based diffusion models. In *CVPR*, 2023. 2
- [9] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 2
- [10] Abhishek Kar, Shubham Tulsiani, João Carneira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015. 2
- [11] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *CVPR*, 2020. 3, 7, 12
- [12] Filippos Kokkinos and Iasonas Kokkinos. To the point: Correspondence-driven monocular 3d category reconstruction. In *NeurIPS*, 2021. 2
- [13] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, 2020. 2, 6, 7, 11, 12
- [14] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *ECCV*, 2020. 2, 6, 7, 12
- [15] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023. 3
- [16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 7, 12
- [17] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023. 2
- [18] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. RealFusion: 360° reconstruction of any object from a single image. In *CVPR*, 2023. 2, 3, 8, 11
- [19] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019. 2
- [20] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 1, 2, 3, 4, 5, 6, 8, 10
- [21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 1, 2
- [22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 4, 10
- [23] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “GrabCut”: interactive foreground extraction using iterated graph cuts. *ACM TOG*, 23(3):309–314, 2004. 12
- [24] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 1, 2, 4, 10
- [25] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 2
- [26] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023. 2
- [27] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023. 2, 3

- [28] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2
- [29] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 2
- [30] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2
- [31] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion. <https://github.com/ashawkey/stable-dreamfusion>, 2022. 10
- [32] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenatorp, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting stable diffusion using cross attention. *arXiv preprint arXiv:2210.04885*, 2022. 12
- [33] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022. 13
- [34] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. DOVE: Learning deformable 3d objects by watching videos. *arXiv preprint arXiv:2107.10844*, 2021. 2
- [35] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. MagicPony: Learning articulated 3d animals in the wild. In *CVPR*, 2023. 2, 3, 6, 7, 11, 12
- [36] Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. De-rendering the world’s revolutionary artefacts. In *CVPR*, 2021. 2
- [37] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *CVPR*, 2020. 2
- [38] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T. Freeman, and Ce Liu. LASR: Learning articulated shape reconstruction from a monocular video. In *CVPR*, 2021. 2
- [39] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. ViSER: Video-specific surface embeddings for articulated 3d shape reconstruction. In *NeurIPS*, 2021. 2
- [40] Gengshan Yang, Minh Vo, Neverova Natalia, Deva Ramanan, Vedaldi Andrea, and Joo Hanbyul. BANMo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022. 2
- [41] Gengshan Yang, Chaoyang Wang, N. Dinesh Reddy, and Deva Ramanan. Reconstructing animatable categories from videos. In *CVPR*, 2023. 2
- [42] Chun-Han Yao, Wei-Chih Hung, Michael Rubinstein, Yuanzhen Lee, Varun Jampani, and Ming-Hsuan Yang. Lassie: Learning articulated shape from sparse image ensemble via 3d part discovery. In *NeurIPS*, 2022. 2, 11, 12

Appendix

This supplementary material contains an extended comparison with prior work (Appendix A.1), a failure case analysis (Appendix A.2), and more technical details (Appendix B), which also include additional results with automatically obtained segmentation masks (Appendix B.2).

A. Additional Results

A.1. Additional Comparisons with Prior Work

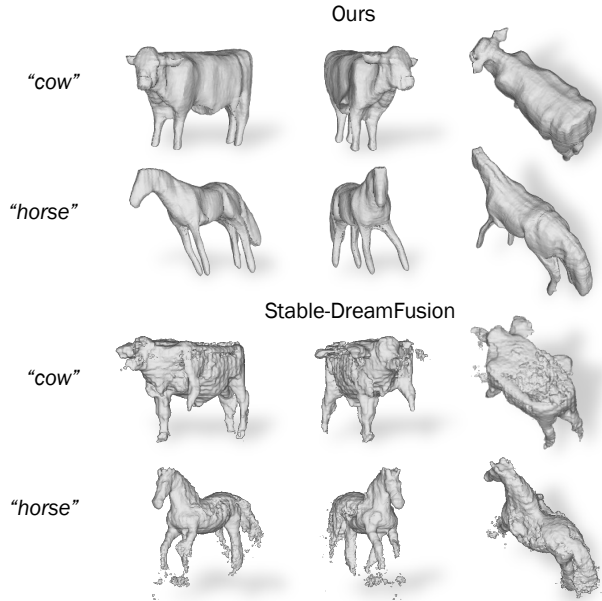


Figure 10: **Comparison with DreamFusion.** We run Stable-DreamFusion [31], an open-source implementation of DreamFusion [20]. Stable-DreamFusion produces shapes with holes and wrong anatomies (for ‘cow’, half of the left front leg is missing; for ‘horse’, there are 5 legs).

DreamFusion To elaborate on how our method can be a preferable candidate for 3D asset generation over category-agnostic methods such as DreamFusion [20], we provide an additional qualitative comparison with Stable-DreamFusion [31], an open-source implementation of DreamFusion that uses Stable Diffusion [22] (rather than the non-public Imagen [24]) as the 2D prior. Given a prompt, for Stable-DreamFusion, we extract the mesh from the *optimised* radiance field, using its implementation out of the box; for our method, we first sample 4 images from Stable Diffusion conditioned on the same prompt and remove the images which are likely to contain truncated instances (using the same automatic pipeline as for pre-processing our virtual training images), and then randomly select one

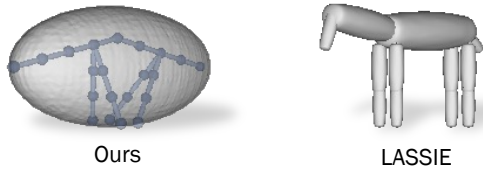


Figure 11: **Comparison of shape initialisation against LASSIE.** Our model begins with a generic ellipsoid with a simple heuristic description of the bone topology, whereas LASSIE [42] initiates with a skeleton featuring bones initialised as cylinders which are reminiscent of the final object structure. LASSIE employs a much stronger hand-crafted initial shape, which simplifies their optimisation process.

image to feed into our category-specific model to get the mesh. We show the generated 3D contents, with their corresponding prompts, in Fig. 10. While Stable-DreamFusion takes 30-60 minutes to produce the mesh, our method uses less than 10 seconds (5 seconds for generating and pre-processing images and 1 second for a single forward pass to predict the shape). In addition, users have more control over the synthesised 3D asset when using our method, as our model can be conditioned on images of the same category (like RealFusion [18]); by contrast, even to get a different mesh with (Stable-)DreamFusion, one has to tweak the prompt, change the random seed, and/or modify the guidance scale and wait for another 30-60 minutes.

LASSIE In addition to our comparison with learning-based methods, we compare with an optimisation-based method, LASSIE [42], both qualitatively and quantitatively, in the task of single-view reconstruction and keypoint transfer respectively. Although our method requires less human input (as shown in Fig. 11), our method produces high-fidelity 3D shapes, while LASSIE interpolates geometric primitive shapes (spheres, cylinders, cones, *etc.*) for each body bone/part, yielding unnatural bone junctions, as demonstrated in Fig. 12. We report the quantitative results of keypoint transfer in Table 2, where we use the standard benchmark of A-CSM [13] for keypoint transfer. Note LASSIE directly *optimises* the reconstruction for each input image, whereas our model has not seen any of these input images during training (nor any real images), showcasing its superb generalisability.

This also highlights the benefit of learning-based methods in unsupervised single-view 3D reconstruction. Given that this problem is highly ill-posed and under-constrained, optimisation-based approaches necessitate strong regularisation, which can limit their ability to generate complex shapes. Conversely, learning-based methods can capitalise

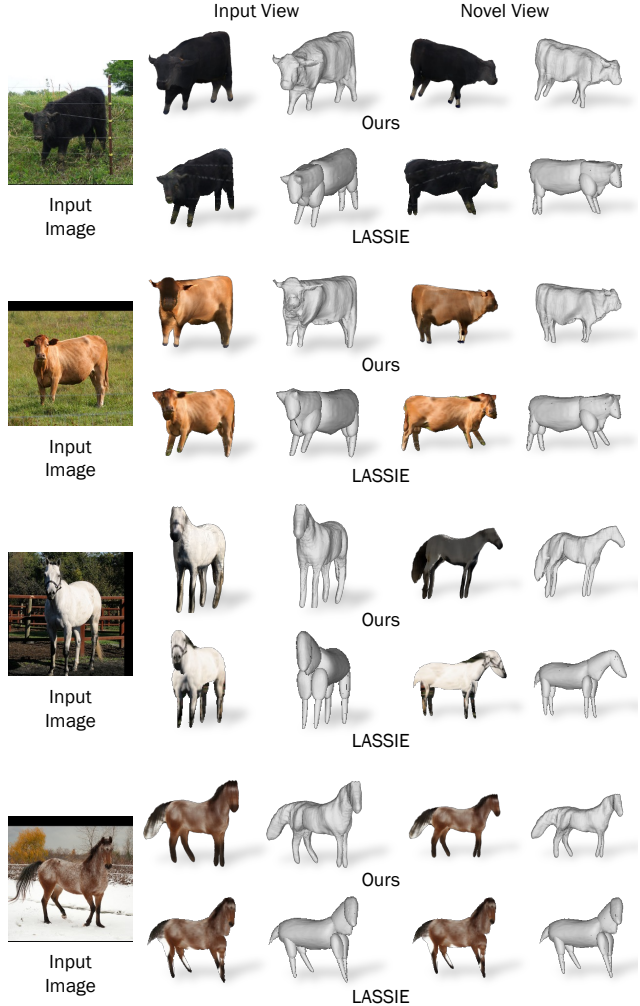


Figure 12: **Comparison with LASSIE.** Our model outputs more realistic 3D shapes than LASSIE [42] without *optimising* the shape on the exact same input images. Note our model disentangles lighting from albedo when predicting appearance, resulting in higher-quality texture when the shape is rendered from a novel view – in the second and third example above, the texture is darker from the unilluminated side.

on multi-view knowledge acquired from their training data. This work demonstrates that one can gain comparable, if not better, multi-view prior knowledge from pre-trained 2D diffusion models, which reduces the need to collect datasets for 3D training and boosts training stability.

A.2. Failure Cases

Similarly to MagicPony [35], the viewpoint prediction is less stable in the case of extreme articulation. As shown in Fig. 13, our method is more likely to predict wrong viewpoints when the animal instance in the input image is facing

Table 2: **Keypoint Transfer on PASCAL VOC – Extended version.** We report PCK@0.1 (\uparrow better). Our method achieves comparable results to MagicPony and outperforms other approaches, without training on any real images. [†]The performance of weakly-supervised methods, such as MagicPony, significantly depends on the cleanliness of the training data as the performance drops when trained on an automatically preprocessed ImageNet subset. *LASSIE [42] directly optimises on the test images; in contrast, our method is not trained on these images, nor on any real images.

Method	Horses	Cows	Sheep
<i>Optimisation-Based Methods</i>			
LASSIE* [42]	0.422	0.375	0.275
<i>Learning-Based Methods</i>			
UMR [14]	0.244	—	—
A-CSM [13]	0.329	0.263	0.286
MagicPony [35]	0.429	0.425	0.262 [†]
Ours (full model)	0.425	0.402	0.328
Ours w/o SDS	0.414	0.370	0.376

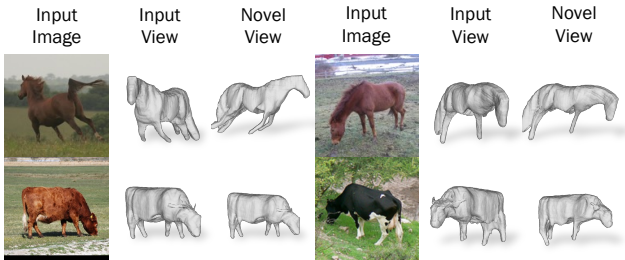


Figure 13: **Failure Cases.** Typical failure modes include incorrect viewpoint prediction and inaccurate articulation.

away from the camera. We hypothesize that this is due to two key reasons. First, Stable Diffusion sometimes generates anatomically incorrect images (examples are shown in Fig. 14), especially when the instance is facing away. Second, even with our engineered prompts, instances facing away from the camera are still underrepresented in the *virtual* training set. In addition, our model does not generalise well to instances with extreme articulations, due to the limited articulation coverage by the *virtual* training data.

B. Additional Technical Details

B.1. Segmentation Masks

After generating the images I , we follow MagicPony and obtain masks using PointRend [11], which is trained on COCO [16]. Unlike MagicPony, instead of focusing on a single category (*i.e.*, horse), our method works on a variety



Figure 14: **Stable Diffusion Failure Cases.** Stable Diffusion can generate anatomically wrong images. We observe that this is typically correlated with which is particularly significant for instances facing away.

of quadruped animal categories, some of which (*i.e.*, pig) do not exist in COCO. In practice, we find PointRend also produces reasonably good masks for pigs when asked to detect its training categories (*e.g.*, cow, elephant, sheep, or horse), despite the semantic mismatch. We, therefore, use the mask with the highest confidence score from detections of these categories.

B.2. Automatic Object Segmentation via Cross-attention

We also test our model on unreal “fictional” objects, such as steampunk cows, generated by Stable Diffusion. Since PointRend cannot detect such objects in general, we adapt a technique based on cross-attention originally developed for interpreting Stable Diffusion [32], and automatically obtain reasonably accurate masks without manual supervision. This method produces attribution heatmaps for a given word in the textual prompt used to generate the image. We condition the generation of the attribution heatmap on the name of the object we are generating (*e.g.* ‘cow’, ‘horse’, ‘sheep’, *etc.*), and threshold the generated heatmap to obtain an initial segmentation map, which is then refined using the classic GrabCut algorithm [23]. This simple unsupervised segmentation method can also replace PointRend in our training pipeline as demonstrated in Fig. 15.

B.3. Removal of Truncated Images

Stable Diffusion sometimes generates portraits or truncated images of an instance, not showing its full body and hence not useful for our purpose of learning a statistical model of an articulated object category. We filter out such images automatically with a simple heuristic: if any of the four borders of a generated image, with a 10-pixel margin, contain more than 25% of masked pixels, we consider the image to be truncated and exclude it from the training dataset.

B.4. Prompt Design

To generate training images using Stable Diffusion, we needed to create an appropriate textual prompt. Starting with a simple prompt, a [V] (where [V] represents the

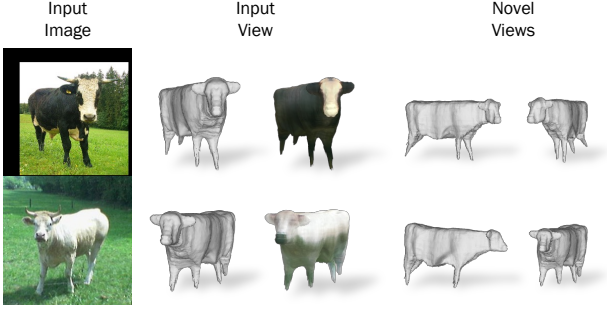


Figure 15: **Qualitative Results Obtained from Automatic Object Segmentation on Cows.** Despite using segmentation masks automatically obtained via cross-attention, our model can still produce 3D shapes that are faithful to the input images and realistic when rendered from novel views.



Figure 16: **Training Images Generated with Stable Diffusion.** Additional examples of our generated training images (after cropping).

word for an animal category, such as horse, cow, or sheep), we iteratively adapted the prompt until we observed a reasonable quality of generated images, displaying good viewpoint coverage. This process took about eight iterations. The final prompt is as follows: a photograph of a single [V], [V] is walking away from the camera. Additionally, we utilise the following negative prompt: wrong anatomy, animal is lying on the ground, lying, dead, black and white photography, human, person. This single prompt, in conjunction with the negative

prompt, generalises effectively across all categories. We show examples of generated images in Fig. 16. Note this engineered prompt is only used for data generation. For SDS gradient computation, we condition Stable Diffusion simply on the prompt a [V] without negative prompts.

B.5. Implementation Details

Here we provide additional implementation details not covered in the main text. We employ the open-source HuggingFace Diffusers library [33] for the Stable Diffusion implementation. We use Stable Diffusion version 1.5, as we observed that version 2.1 more frequently results in incorrect anatomies. When generating training images with Stable Diffusion, we set the guidance scale to 10 and perform 50 inference steps using half-precision floating point. The SDS loss used during training, Eq. (2) in the main text, is weighted with 1×10^{-4} . The complete training takes 20 hours for 120 iterations on a single Nvidia A40 GPU.