STAG: Smart Tools and Applications in Graphics (2025)
M. Comino Trinidad, C. Mancinelli, D. Cabiddu, and D. Giorgi (Editors)

Combining CNN Feature Extraction and Kolmogorov-Arnold Networks Regression for Procedural 3D Shape Generation

G. Manfredi[1], N.Capece[1], U. Erra[1], and A. Gruosso[1]

[1]University of Basilicata, Department of Engineering, Italy

Abstract

Generating 3D objects with complex, nonlinear shapes directly from images is still an open research area. To address this problem, several state-of-the-art methods use Deep Learning (DL) to predict a set of parameters from images, which are then used to generate the 3D geometry, leveraging the characteristics of procedural modeling. Recently, Kolmogorov-Arnold Networks (KANs) have emerged as an alternative to traditional Multilayer Perceptrons (MLPs) in DL, and have been successfully integrated into architectures such as Convolutional Neural Networks (CNNs), Graph Neural Networks, and Transformers. In this work, we propose a DL architecture consisting of a hybrid CNN-KAN network for parametric 3D model generation from images. The model combines the ability of KANs to capture complex nonlinear functions with the strong visual feature extraction capabilities of CNNs. The method is evaluated using both quantitative error metrics and qualitative visualizations comparing predicted shapes with ground truth, and its performance is compared against a more standard CNN-MLP architecture.

CCS Concepts

• ***Computing methodologies*** → ***Artificial intelligence; Modeling methodologies;***

1. Introduction

The creation of 3D content, especially 3D objects, has become of central importance in different fields from game development to simulation. As these fields continue to grow, there is an increasing demand for methods that can automatically synthesize 3D objects while minimizing human input. A common approach for generating large collections of shapes is procedural modeling, where an object's shape is defined by a set of interpretable parameters. This offers precision, repeatability, and flexibility, but it also introduces challenges as even skilled users must navigate a large parameter space to achieve the desired geometry [HSvK25]. By contrast, obtaining images of the object is often easier and more intuitive than manually adjusting parameters or modeling geometry from scratch [MCEG23]. However, inferring the correct parameters from image data is non-trivial. Procedural models are often non-differentiable or non-invertible, making the inverse mapping from pixels to parameters arduous [HSvK25]. In literature, several works have approached this task using a combination of CNNs and MLPs, where the CNN serves as a feature extractor from images and the MLP performs the regression of the procedural parameters [GFK*18; MCEG23; CCY*25], exploiting the complementary strengths of the two architectures. However, traditional MLPs often struggle to approximate highly nonlinear mappings without requiring deep or wide architectures, which can lead to overfitting and poor generalization. Recently, Kolmogorov–Arnold Networks (KANs) have been proposed as an alternative to MLPs. As demonstrated by Liu et. al. [LWV*24], by replacing scalar weights with learnable univariate functions, KANs can significantly enhance the network's ability to capture complex nonlinear relationships while maintaining parameter efficiency. This makes them particularly well-suited for tasks such as procedural parameter regression, where the mapping from visual features to geometric parameters is highly nonlinear and difficult to model with standard MLPs.

In this work, we propose a DL architecture characterized by a backbone CNN designed to extract features from a single input image. These features are then sent to a set of KAN layers, which predict the values of the parameters corresponding to the shape represented in the input image. The proposed architecture has been trained using a supervised learning approach with a synthetically generated dataset comprising pairs of rendered images of 3D shapes and dictionaries of the corresponding parameter values. The shapes were defined through a procedural formulation that combines a helicoidal torsion with a sinusoidally modulated radius, resulting in spiral tube–like structures. This representation is particularly advantageous, as it requires only a small set of parameters while still producing geometries that are diverse and structurally coherent, making it well-suited for regression-based prediction tasks. The method is assessed quantitatively using error metrics and qualitatively through visual comparisons between predicted and ground-truth shapes, with performance benchmarked against a standard CNN–MLP baseline. Experimental results demonstrate that KAN-based regression, particularly with the FourierKAN vari-

DOI: 10.2312/stag.20251327

ant, achieves more expressive parameter estimation despite a modest increase in model size, leading to more accurate and consistent 3D reconstructions. The contributions of this paper are:

- A hybrid CNN–KAN architecture that combines pretrained CNNs for visual feature extraction with Kolmogorov–Arnold Networks (KANs) for regressing parameters of parametric 3D shapes from a single image.
- A synthetic multi-view dataset generated in Blender, consisting of paired images and parameters to support effective training and validation.
- Comprehensive evaluation, both quantitative and qualitative, demonstrating consistent improvements in 3D reconstruction accuracy and visual fidelity.

Beyond the methodological contribution, the proposed framework has several promising applications. It could be employed in the reverse engineering of CAD models from 2D images, enabling the recovery of compact, editable parametric representations. In the cultural heritage domain, it could support the digital reconstruction of architectural details from limited photographic data. In the gaming and entertainment industries, the approach could automate the creation of 3D assets from concept art or reference images, reducing manual modeling time and streamlining content creation for virtual environments, films, and simulations. The remainder of this paper is structured as follows: Section 2 reviews related work on 3D reconstruction and parametric modeling; Section 3 provides background on KANs and their variants; Section 4 describes the synthetic dataset used for training and evaluation; Section 5 details the proposed CNN–KAN architecture and training procedure; Section 6 presents quantitative and qualitative results, including comparisons between MLP and KAN-based regression heads. Finally, Section 7 summarizes our findings and discusses directions for future work.

The full implementation, along with dataset generation scripts and pretrained model weights, is available on GitHub at: **https://github.com/antoniogrs/kan_parametric_reconstruction**.

2. Related Work

In this work, we address the problem of parametric surface estimation from single images, aiming to recover 3D parametric models from a single 2D observation. Recent works increasingly focus on deep learning approaches for generating 3D models using procedural modeling methods [DVH*22; RLM*23]. In particular, several studies have explored parametric surface estimation directly from single images or point clouds. For instance, Gao *et al.* [GTG*19] proposed a deep learning architecture for reconstructing spline-based curves and surfaces from images or point clouds, while Sharma *et al.* [SLK*20] focused on decomposing 3D point clouds into parametric surface patches, including B-spline primitives. Smirnov *et al.* [SFK*20] extended this line by predicting parametric shape primitives from noisy geometric data using distance fields. Other works focused on related problems of surface fitting from point clouds, such as Ben-Shabat and Gould [BG20], which learns local polynomial surfaces via neural-weighted least squares. More recent approaches leverage UNet-based architectures for robust B-spline surface parameterization from point clouds [TY25], and transformer-based networks for viewpoint-invariant 3D shape generation from single images using differentiable NURBS surfaces [YLY21].

Wang *et al.*[WCPM18] employ a multi-encoder-decoder network to reconstruct garments, jointly predicting 2D patterns and body shape. Similarly, Unlu *et al.* [USB22] design two networks to infer silhouettes and joints from mannequin sketches, which are then used for 3D reconstruction. Early methods were mostly rule-based, for instance, using sketches to compute procedural parameters [HKYM17] or relying on genetic algorithms [HSS17]. Dai *et al.* [DZL23] highlighted the potential of rule-based methods as a way to produce dynamic, parametrized 3D models for Digital Twins that are easier to adapt and synchronize with their physical counterparts, overcoming photogrammetry or CAD conventional workflows that often require domain expertise and are hard to update [NSEL22]. Procedural modeling has also been applied to 3D materials; a recent framework [TRT*22] indeed, can predict parameters for procedural material models from images, enabling infinite variations of high-resolution, photo-realistic materials. Moreover, Large and Vision-Language Models have been applied to procedural modeling, enabling users to generate 3D content from natural language descriptions and providing greater flexibility and control over scene design [FZF*23; SHD*25; ZWH*24].

In practical pattern recognition applications, input data are often featured in high-dimensional spaces. Compared to classical MLPs, KANs can represent complex functions with fewer parameters, improving efficiency and interpretability. Although MLPs are typically less computationally expensive than KAN under the same conditions, the use of functions instead of linear parameters allows KANs to achieve better results with much shallower neural networks. Consequently, in high-dimensional contexts, KANs can be significantly more efficient than traditional MLPs [YYW24]. For these reasons, KANs were already used for various computer vision tasks, including: image classifications, *e.g.*,outperforming baseline models with interesting accuracy on MNIST and SVHN datasets as shown by Ferdaus *et al.* [FAI*25]; semantic segmentation, where Ma *et al.* [MWH*25] improved the decoder's ability to capture fine-grained details of their network during decoding; and person and vehicle re-identification. In the latter, Zhan *et al.* [ZSL*25] integrated KANs into the attention module, the KAN Spatial-Channel Merge Attention, which enabled a residual network to better separate spatial channel features, showing potential improvement over existing methods. KANs have also been applied to AI-generated image analysis. For example, Anon and Emon [AE24] proposed a hybrid model combining semantic embeddings with an MLP containing KAN layers to classify real versus AI-generated images. This model was able to capture subtle details previously ignored, outperforming a standard MLP on out-of-distribution tests. While Generative Adversarial Networks (GANs) enhanced with contrastive learning have been used for feature generation, limitations such as randomness in feature selection and the limited expressiveness of MLP-based projection heads often result in suboptimal performance. To address this, Mahara *et al.* [MRWS24] introduced KAN-Attn, which incorporates an attention-based query selection mechanism to choose relevant features during contrastive learning. KAN-Attn represents one of the first applications of Kolmogorov–Arnold Networks to enhance feature generation, improv-

ing expressiveness in map generation tasks. KANs have also been successfully applied in other generative contexts, such as genomic sequence generation [CP25] and graph modeling [LZWX25]. In each case, the underlying idea is the same: leverage the flexible functional representation of KANs to produce new data or transformations more effectively than traditional networks.

To the best of our knowledge, KANs have not yet been applied to 3D content generation, including parametric modeling. Existing approaches for predicting procedural parameters typically rely on MLPs or other deep neural architectures, which often require very deep networks to handle high-dimensional input and capture complex nonlinear relationships. By leveraging KANs, our work introduces a novel methodology for 3D procedural parameter prediction that combines the benefits of KANs with the expressiveness required for complex 3D structures. This allows us to predict procedural parameters accurately and provides a new scientific pathway for combining high-dimensional function approximation with 3D procedural modeling.

3. Background: Kolmogorov–Arnold Networks

KANs are a neural architecture inspired by the Kolmogorov–Arnold representation theorem [Kol61], which states that any multivariate continuous function can be expressed as a finite composition of continuous univariate functions and addition. This theoretical foundation motivates KANs as a function-oriented alternative to conventional MLPs. Like MLPs, KANs adopt a fully connected structure, and it is based on the idea of stacking layers to build deeper networks and achieve increasingly expressive representations. A KAN layer with input dimension n_{in} and output dimension n_{out} can be expressed as a matrix of univariate functions:

$$\Phi = \{\phi_{q;p}\}; \quad p = 1,2,\ldots,n_{in}; \quad q = 1,2,\ldots,n_{out}, \qquad (1)$$

where each function $\phi_{q;p}$ has trainable parameters. By stacking multiple KAN layers, one can build deep architectures capable of representing complex functional relationships. The effectiveness of KANs depends largely on the choice of the underlying univariate functions. Liu *et al.*[LWV*24] introduced splines as the basic univariate building blocks. Splines are piecewise polynomial functions with guaranteed continuity up to a chosen derivative order at the junctions (knots). They provide a continuous and differentiable representation of functions from numerical data while preserving a local structure: a change in a coefficient or a knot only affects a limited portion of the function. This property significantly improves numerical efficiency and generalization ability. The first reference implementation, `pykan`, demonstrated these advantages, particularly in the context of symbolic regression. However, its use outside this domain remained limited due to architectural constraints and slow training times. Shortly after Liu *et al.*'s publication, the developer Blealtan released `efficient-kan` [Ble24], specifically addressing the performance bottlenecks of `pykan`. In the original implementation, activation functions required expanding the input tensor to shape $(\text{batch\_size}, n_{out}, n_{in})$, which caused large memory overhead. Since the activation functions are linear combinations of a fixed set of B-splines, `efficient-kan` reformulated the computation as applying the input to a set of basis functions followed by a linear combination of the results. This reduces memory consumption and expresses the operation as efficient matrix multiplications suitable for both forward and backward passes. `efficient-kan` also introduced a more practical sparsification strategy. Whereas Liu *et al.*applied L1 regularization on sampled inputs (computationally expensive), `efficient-kan` instead applies L1 regularization directly on the weights, a standard and efficient approach in neural networks, while retaining interpretability. In fact, `efficient-kan` has influenced the structure and development of later implementations more strongly than the more rigorous yet less accessible implementation of Liu *et al.*. The first to explicitly follow Liu's suggestion of exploring alternative univariate functions was Ziyao Li [Li24], who introduced Gaussian Radial Basis Functions (RBFs). In his work [Li24], he proposed the theoretical foundations and later implemented them in `fast-kan` [Li25], which leverages Gaussian RBF kernels together with a layer normalization step to prevent inputs from drifting outside the RBF domain. Building on these developments, Zhang *et al.*[ZFCW25b; ZFCW25a] proposed the Kolmogorov–Arnold Fourier Network (KAFN), designed to overcome the limitations of KANs in high-dimensional settings. KAFNs incorporate trainable Random Fourier Features together with a hybrid GELU–Fourier activation function. This hybrid mechanism utilizes the non-linearity of GELU while dynamically adjusting the Fourier components: during early training, it emphasizes low-frequency structures and gradually increases frequency resolution as optimization progresses. As a result, KAFs achieve strong generalization with a contained number of parameters, making them a powerful extension of KANs for broader applications.

4. Dataset

As introduced in Section 1, our objective is to generate parametric 3D shapes from a single image using a hybrid CNN-KAN architecture. This network is designed to predict the values of a set of parameters that define the shape depicted in the input image. Unlike other 3D reconstruction tasks that focus on directly reconstructing a mesh or point cloud, our approach involves learning the underlying generating function for the shape. This characteristic makes our problem particularly well-suited to KANs, as their attention mechanism can effectively capture the functional relationships between parameters and geometry. One of the main challenges of supervised learning, however, lies in the need to construct large labeled datasets that are not only extensive but also carefully structured to achieve reliable results. To address this, we created a dedicated synthetic dataset using Blender [Fou25], a 3D modeling software that can be scripted in Python. This capability allowed us to develop a script that automatically generates pairs of rendered images of the parametric shapes along with dictionaries containing the corresponding parameter values, thus providing well-structured ground-truth data for training and evaluation.

The parametric 3D shape chosen for dataset generation is a spiral tube–like surface defined through a procedural formulation that combines a helicoidal torsion with a sinusoidally modulated radius. This choice provides a good balance between variability and simplicity, as the model can generate a wide range of distinct geometries while keeping the parameter space compact and interpretable.

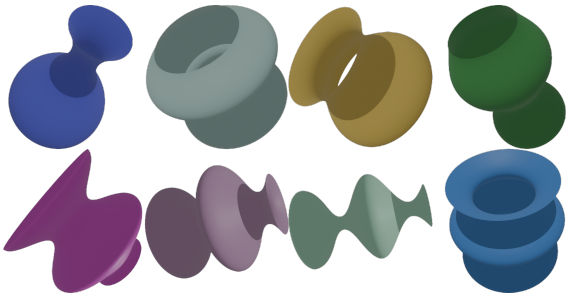

Figure 1: *Examples of images rendered from the parametric formulation of the spiral tube. Notice the variability across objects due to different torsion and waviness parameters.*

The formulation requires only a small set of parameters yet produces shapes that are both diverse and structurally coherent. The explicit mathematical definition of the surface is given by Equation 2:

$$\begin{cases} x(u,v) = \big[r + A_w \cos(f_w v)\big] \cos\big(u + A_t \sin(f_t v)\big), \\ y(u,v) = \big[r + A_w \cos(f_w v)\big] \sin\big(u + A_t \sin(f_t v)\big), \\ z(u,v) = v, \end{cases} \tag{2}$$

where $u \in [0, 2\pi)$ and $v \in [0, h]$ are the angular and vertical parameters of the surface domain, respectively; r is the *base radius* of the structure; A_w and f_w control the *amplitude* and *frequency* of the radial wavy modulation along the vertical axis; A_t and f_t define the *amplitude* and *frequency* of the helicoidal torsion applied around the central axis; $z(u,v) = v$ ensures a vertical elongation of the surface along the height h. This parametric formulation generates a family of column-like spiral structures characterized by a sinusoidally modulated radius and a helicoidal twist, resulting in diverse yet structurally coherent 3D geometries. The variability introduced by the torsion and radial modulation is clearly visible in Figure 1, which shows examples of images generated using the parametric formulation of the spiral tube.

To enable the network to infer 3D structure effectively, each object is rendered from multiple viewpoints, simulating the way a human observer examines an object from different angles. Cameras are positioned at predefined angles around the object and can be randomly perturbed along the horizontal plane to increase variability. For each rendered view, the corresponding parameter values are saved in a `.json` file, providing the ground-truth labels used for supervised training.

To further enhance the diversity and realism of the dataset, additional elements into the Blender scene have been introduced. First, a dynamic lighting system was implemented by attaching a sunlight source to the moving camera. This ensured that each rendered view contained informative shadows, which are useful for perceiving the object's geometry, while avoiding excessive noise caused by poorly illuminated regions. In addition, randomized colors were assigned to the objects so that each rendered instance differed not only in shape but also in appearance. This variability provides the network with richer visual cues, with the aim of helping it generalize better during training.

Following the idea of Manfredi *et al.*[MCEG23], each object has been rendered in five views, with four reserved for training and one for validation. During testing, all five views are used to ensure a comprehensive evaluation of the network's ability to generalize across viewpoints. The final dataset comprises $11,824$ training images, $2,956$ validation images, and $1,000$ test images, all at a resolution of 618×618 pixels, with each image paired with its corresponding parameter `.json` file containing the `radius`, `height`, `twist_amp`, `twist_freq`, `wave_amp`, and `wave_freq` parameters' values.

5. CNN-KAN Architecture

The proposed architecture is characterized by a backbone CNN that processes reshaped input images of size 224×224. This choice of resolution lies within the typical range for CNN-based training tasks (64×64 to 256×256), where lower resolutions generally improve training efficiency while preserving sufficient visual detail for robust feature extraction. Since our approach follows a supervised learning paradigm, each input image is paired with the corresponding set of parameters that describe the underlying 3D shape. These parameters are organized into a vector of size $1 \times n_p$, with $n_p = 6$. Before training, parameter values are normalized using Min–Max Scaling to ensure balanced learning across dimensions. In this work, our primary objective was to isolate and evaluate the expressivity of KAN layers within a controlled experimental setting. To ensure that performance differences could be attributed primarily to the regression head rather than to variations in feature extraction, we deliberately employed lightweight and well-established CNN backbones. This design choice allows for a fair and interpretable comparison between traditional MLP and KAN-based regression architectures, minimizing confounding factors related to backbone capacity or training complexity. The CNN backbone is responsible for extracting meaningful features from the input images. For this purpose, we selected SqueezeNet v1.1 [IHM*16] with pretrained ImageNet weights. SqueezeNet was selected mainly for its efficiency, as despite its small size, it offers representational capacity comparable to larger models. Before adopting pretrained networks, we also experimented with a custom lightweight CNN composed of three convolutional blocks followed by four FastKAF layers [ZFCW25a]. However, this configuration yielded low accuracy, indicating that the network lacked sufficient expressive power for the task. EfficientNet [TL19], a state-of-the-art architecture known for its efficiency and scalability, was also tested, but it exhibited significant overfitting even when regularization techniques such as dropout were applied. In contrast, SqueezeNet achieved a more favorable balance between expressiveness and generalization, making it the most suitable backbone for our architecture. The output of the final convolutional layer of the CNN is flattened and passed into a sequence of KANs. In particular, we adopted the FastKAF layer implementation, which is based on KAFs. This choice was motivated by the ability of KAF layers to efficiently approximate complex functional relationships through learnable Random Fourier Features, while retaining good generalization with a reduced number of parameters. Our

KAN component is structured as a sequence of five fully connected FastKAF layers, progressively reducing the feature dimensionality from the CNN backbone to the final parameter prediction. Specifically, the first layer maps the 512-dimensional input vector produced by SqueezeNet to 256 features, followed by a second layer that reduces the dimensionality to 128. The third and fourth layers further compress the representation to 64 and 32 dimensions, respectively. Finally, the fifth layer outputs the six target parameters that define the procedural 3D shape. Each of the first four layers is followed by a dropout operation with a probability of 0.6 to mitigate overfitting, while the last layer directly produces the regression output. Unlike traditional feedforward networks, no additional nonlinear activation functions are applied between KAN layers, since their activation behavior is inherently captured by the learnable univariate functions of the KAF formulation. From an architectural perspective, the number of KAN layers and neurons per layer are the primary hyperparameters influencing the model's expressivity. Experimental tests confirmed that reducing the number of KAN layers leads to a measurable decrease in regression accuracy, as the network becomes less capable of modeling complex nonlinear relationships. Conversely, increasing the number of KAN layers enhances expressivity but also results in higher computational cost and a greater tendency toward overfitting. Beyond depth and width, several FastKAF-specific hyperparameters also play a critical role in expressivity and stability. In particular, we set the grid size of the RBFs in FastKAF layers to 8, which provides sufficient flexibility for accurately approximating the univariate functions required by the KAN framework, balancing functional expressivity with computational efficiency. An overview of the complete CNN–KAN architecture is illustrated in Figure 2. For training, we adopted the Mean Squared Error (MSE) loss, a common choice for regression tasks. Weight optimization was performed using the AdamW [LH17] optimizer with a learning rate of 5×10^{-6}. Compared to standard Adam, AdamW improves generalization by decoupling weight decay from the gradient update, making it particularly effective in settings where overfitting is a concern. Additionally, an early stopping criterion based on validation loss was optionally applied using a patience threshold, preventing unnecessary epochs once the network ceased to improve. All experiments were conducted on an Alienware Aurora R16 workstation, equipped with an Intel64 Family 6 Model 183 CPU running at 2.1 GHz, 16 GB of RAM, and an NVIDIA GeForce RTX 4060 Ti GPU with 8 GB of GDDR6 VRAM, running CUDA 12.9. The system operated under Windows 11. The software environment included Python 3.13.2, PyTorch 2.8.0, TorchVision 0.22.0, and TorchAudio 2.6.0. To complete 200 epochs of training, the CNN–KAN architecture required approximately 3 hours, while the inference time per image was around 85 milliseconds.

6. Results

To evaluate the performance of our models, we considered both traditional regression metrics and geometric fidelity measures. Specifically, we computed the coefficient of determination R^2, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the Hausdorff distance (HD) [CRS98] between ground-truth and reconstructed 3D models of the test set, normalized in the unit sphere.

We compared the three different CNN backbones mentioned in Section 5: the simple CNN with three convolutional blocks, EfficientNet-B0, and SqueezeNet v1.1, each coupled first with traditional MLP layers and then with FastKAF-based KAN layers. For EfficientNet-B0 and SqueezeNet, dropout was applied to mitigate or eliminate overfitting, respectively, while the simpler CNN did not require dropout. Table 1 shows that replacing MLP layers with KAN layers generally improved performance across all backbones, with larger gains observed for EfficientNet-B0 compared to the Simple CNN. Among the tested configurations, SqueezeNet combined with KAN layers achieved the best overall results, obtaining the highest R^2 and the lowest MSE, RMSE, and Hausdorff distance. This demonstrates its superior ability to both predict accurate parameters and reconstruct faithful 3D geometries.

Table 1: *Quantitative evaluation of different CNN backbones combined with MLP and KAN regression heads. Metrics include coefficient of determination (R^2), mean squared error (MSE), root mean squared error (RMSE), and Hausdorff distance (HD). Higher R^2 and lower errors indicate better performance.*

Architecture	$R^2 \uparrow$	MSE $\downarrow$	RMSE $\downarrow$	HD $\downarrow$
Simple CNN + MLP	0.035	0.082	0.286	0.608
Simple CNN + KAN	0.066	0.079	0.281	0.616
EfficientNet-B0 + MLP	0.327	0.058	0.240	0.351
EfficientNet-B0 + KAN	0.507	**0.042**	**0.205**	0.647
SqueezeNet + MLP	0.493	0.043	0.208	0.417
SqueezeNet + KAN	**0.506**	**0.042**	**0.205**	**0.335**

For qualitative evaluation, we compared the reconstructed meshes against their corresponding ground-truth models to assess how accurately the predicted parameters capture the 3D geometry. Figure 3 presents three representative cases (best, mean, and worst) selected based on the Hausdorff distance of the SqueezeNet + KAN model. Each case compares the reconstructions obtained from SqueezeNet + MLP, SqueezeNet + KAN, and EfficientNet-B0 + KAN. The Hausdorff distance was computed on meshes normalized within the unit sphere, ensuring scale-independent evaluation across all models. In the error maps, vertices are color-coded using an absolute colormap scale, allowing direct visual comparison of reconstruction quality between architectures. Across all scenarios, SqueezeNet + KAN consistently achieves lower reconstruction errors and visually smoother surfaces compared to SqueezeNet + MLP, confirming that KAN layers enhance the expressive capacity of compact backbones by capturing fine-grained geometric details more effectively. While EfficientNet-B0 + KAN achieves competitive quantitative results (Table 1), its qualitative reconstructions are sometimes less stable, indicating that lower numerical errors do not always correspond to perceptually superior shapes and highlighting the importance of combining quantitative and qualitative assessments.

7. Conclusion

In this work, we addressed the problem of parametric regression for procedural 3D shape reconstruction from images. After exploring different combinations of backbones and regression heads, we selected SqueezeNet with FastKAF layers as the reference

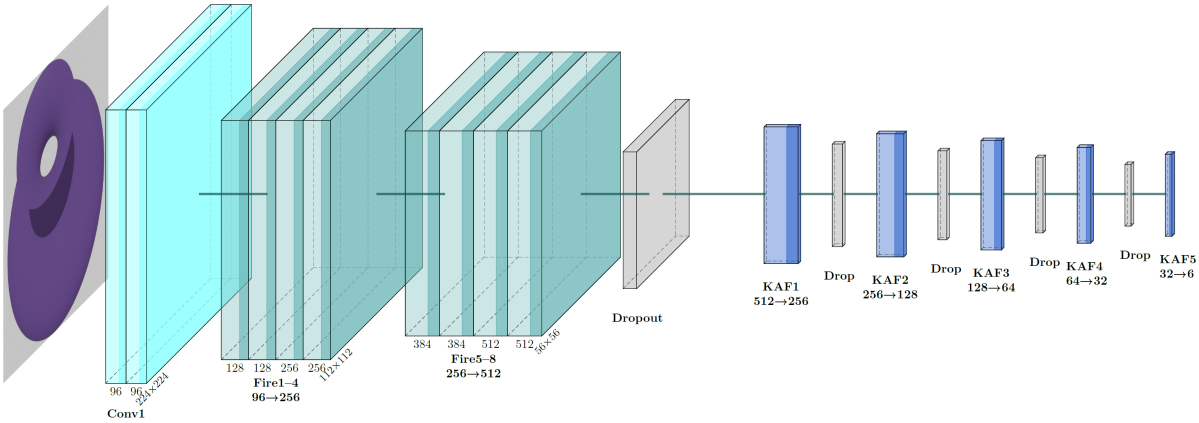

Figure 2: *Overview of the proposed CNN–KAN architecture. The backbone CNN (SqueezeNet v1.1) extracts compact visual features from the input image, which are then passed through a sequence of FastKAF layers. These progressively reduce the feature dimensionality and ultimately predict the six parameters defining the procedural 3D shape.*

architecture. This choice was motivated by its balance of efficiency, compactness, and strong representational capacity, making it well-suited for the task of predicting procedural parameters. We compared this configuration against alternative setups, including a simple CNN and EfficientNet-B0, each tested with both MLP and KAN regression heads. Quantitative evaluation was carried out using R^2, MAE, RMSE, and Hausdorff distance, while qualitative evaluation was performed through visual comparison of reconstructed meshes against ground-truth models. Our experiments showed that replacing MLP layers with KAN layers consistently improved performance across architectures, demonstrating the effectiveness of function-oriented representations for regression tasks. Among the tested configurations, SqueezeNet combined with KAN layers achieved the best overall performance, striking a balance between compactness, generalization, and accuracy. While EfficientNet-B0 with KAN layers performed competitively in terms of numerical metrics, its reconstructions were qualitatively closer to those obtained with the less expressive SqueezeNet+MLP, highlighting the importance of evaluating both quantitative and qualitative aspects of the task. A central goal of this study was to isolate and evaluate the expressivity of KAN layers within a controlled experimental setting. To ensure that observed performance differences could be attributed primarily to the regression head rather than to variations in image feature extraction, we deliberately employed lightweight and well-established CNN backbones, such as SqueezeNet and EfficientNet-B0. This approach provided a fair and interpretable comparison between traditional MLP and KAN-based regression architectures, minimizing confounding effects related to backbone capacity or training complexity. Nevertheless, integrating modern encoders, such as Vision Transformers [Dos20] or foundation models like DINOv3 [SVS*25], represents a promising avenue for future research. Overall, the integration of KANs into CNN-based regression pipelines has proven to be a promising direction for parametric 3D modeling. The results suggest that even lightweight backbones, when equipped with expressive KAN layers, can deliver accurate and reliable reconstructions. Future work will focus on increasing dataset complexity by introducing models with more parameters and intricate geometries to test the limits of KANs against traditional architectures. Improving the computational efficiency of KANs remains another priority, for example, through pruning strategies and methods for extracting and analyzing learned functions. These improvements could not only enhance scalability but also provide users with interpretable outputs such as visualizations of learned functions and symbolic formulas. Finally, exploring KAN-based convolutional backbones in place of pretrained CNNs may open the door to architectures that are both highly expressive and function-oriented from end to end.

References

[AE24] ANON, TAHARIM RAHMAN and EMON, JAKARIA ISLAM. "Detecting the undetectable: Combining kolmogorov-arnold networks and mlp for ai-generated image detection". *arXiv preprint arXiv:2408.09371* (2024) 2.

[BG20] BEN-SHABAT, YIZHAK and GOULD, STEPHEN. "DeepFit: 3D Surface Fitting via Neural Network Weighted Least Squares". Glasgow, United Kingdom: Springer-Verlag, 2020, 20–34. ISBN: 978-3-030-58451-1. DOI: 10.1007/978-3-030-58452-8_2. URL: https://doi.org/10.1007/978-3-030-58452-8_2 2.

[Ble24] BLEALTAN. *efficient-kan*. https://github.com/Blealtan/efficient-kan, last accessed on 2025-09-23. 2024 3.

[CCY*25] CHEN, ZHIHONG, CHEN, XUEYUN, YE, CHENGHONG, et al. "Neural radiance fields assisted by image features for UAV scene reconstruction". *Scientific Reports* 15.1 (2025), 30608 1.

[CP25] CHEREDNICHENKO, OLEKSANDR and POPTSOVA, MARIA. "Kolmogorov–Arnold networks for genomic tasks". *Briefings in Bioinformatics* 26.2 (2025), bbaf129 3.

[CRS98] CIGNONI, PAOLO, ROCCHINI, CLAUDIO, and SCOPIGNO, ROBERTO. "Metro: measuring error on simplified surfaces". *Computer graphics forum*. Vol. 17. 2. Wiley Online Library. 1998, 167–174 5.

[Dos20] DOSOVITSKIY, ALEXEY. "An image is worth 16x16 words: Transformers for image recognition at scale". *arXiv preprint arXiv:2010.11929* (2020) 6.

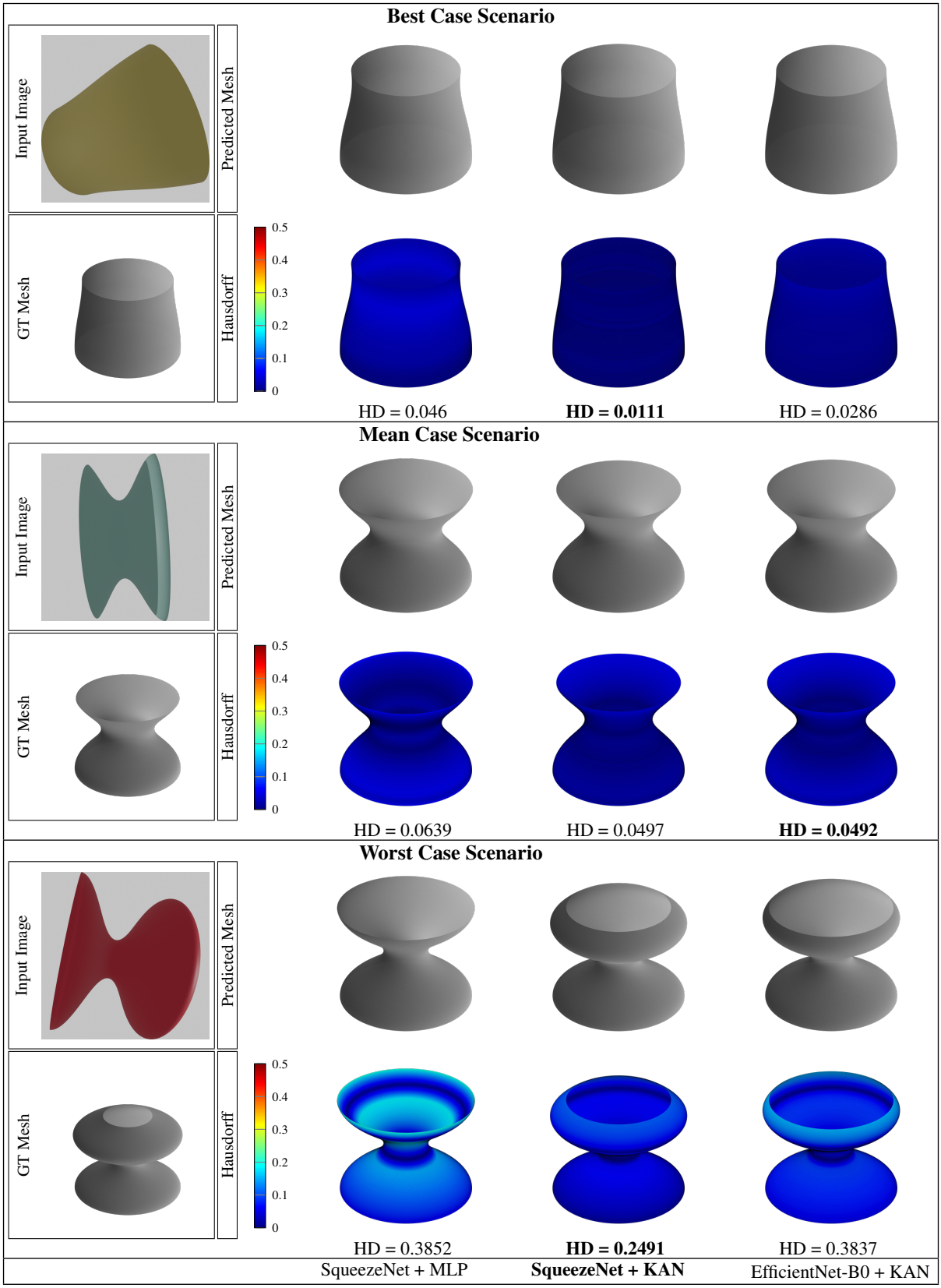

Figure 3: *Qualitative comparison of 3D reconstructions for the best, mean, and worst case scenario of SqueezeNet+KAN. The **Hausdorff** rows color vertices by the absolute Hausdorff distance to indicate reconstruction errors.*

[DVH*22] DEITKE, MATT, VANDER BILT, ELI, HERRASTI, ALVARO, et al. "ProcTHOR: large-scale embodied AI using procedural generation". *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088 2.

[DZL23] DAI, YAQI, ZHONG, RAY Y., and LAU, HENRY Y.K. "Procedural content generation method for creating 3D graphic assets in Digital Twin". *Cogent Engineering* 10.1 (2023), 2216859. DOI: 10.1080/23311916.2023.2216859 2.

[FAI*25] FERDAUS, MD MEFTAHUL, ABDELGUERFI, MAHDI, IOUP, ELIAS, et al. "KANICE: Kolmogorov-Arnold Networks with Interactive Convolutional Elements". *Proceedings of the 4th International Conference on AI-ML Systems*. AIMLSystems '24. New York, NY, USA: Association for Computing Machinery, 2025. ISBN: 9798400711619. DOI: 10.1145/3703412.3703427 2.

[Fou25] FOUNDATION, BLENDER. *Blender*. https://www.blender.org/, last accessed on 2025-09-23. 2025 3.

[FZF*23] FENG, WEIXI, ZHU, WANRONG, FU, TSU-JUI, et al. "LayoutGPT: compositional visual planning and generation with large language models". *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc., 2023 2.

[GFK*18] GROUEIX, THIBAULT, FISHER, MATTHEW, KIM, VLADIMIR G, et al. "A papier-mâché approach to learning 3d surface generation". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 216–224 1.

[GTG*19] GAO, JUN, TANG, CHENGCHENG, GANAPATHI-SUBRAMANIAN, VIGNESH, et al. "Deepspline: Data-driven reconstruction of parametric curves and surfaces". *arXiv preprint arXiv:1901.03781* (2019) 2.

[HKYM17] HUANG, H., KALOGERAKIS, E., YUMER, E., and MECH, R. "Shape Synthesis from Sketches via Procedural Models and Convolutional Networks". *IEEE Transactions on Visualization and Computer Graphics* 23.8 (2017), 2003–2013. DOI: 10.1109/TVCG.2016.2597830 2.

[HSS17] HAUBENWALLNER, K., SEIDEL, H.-P.R, and STEINBERGER, M. "ShapeGenetics: Using Genetic Algorithms for Procedural Modeling". *Comput. Graph. Forum* 36.2 (May 2017), 213–223. ISSN: 0167-7055. DOI: 10.1111/cgf.13120. URL: https://doi.org/10.1111/cgf.13120 2.

[HSvK25] HOSSAIN, ISHTIAQUE, SHEN, I-CHAO, and van KAICK, OLIVER. "Approximating Procedural Models of 3D Shapes with Neural Networks". *Computer Graphics Forum*. Wiley Online Library. 2025, e70024 1.

[IHM*16] IANDOLA, FORREST N, HAN, SONG, MOSKEWICZ, MATTHEW W, et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size". *arXiv preprint arXiv:1602.07360* (2016) 4.

[Kol61] KOLMOGOROV, A. NIKOLAEVICH. *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society, 1961 3.

[LH17] LOSHCHILOV, ILYA and HUTTER, FRANK. "Decoupled weight decay regularization". *arXiv preprint arXiv:1711.05101* (2017) 5.

[Li24] LI, ZIYAO. "Kolmogorov-arnold networks are radial basis function networks". *arXiv preprint arXiv:2405.06721* (2024) 3.

[Li25] LI, ZIYAO. *fast-kan*. https://github.com/ZiyaoLi/fast-kan, last accessed on 2025-09-23. 2025 3.

[LWV*24] LIU, ZIMING, WANG, YIXUAN, VAIDYA, SACHIN, et al. "Kan: Kolmogorov-arnold networks". *arXiv preprint arXiv:2404.19756* (2024) 1, 3.

[LZWX25] LI, LONGLONG, ZHANG, YIPENG, WANG, GUANGHUI, and XIA, KELIN. "Kolmogorov–Arnold graph neural networks for molecular property prediction". *Nature Machine Intelligence* (2025), 1–9 3.

[MCEG23] MANFREDI, GILDA, CAPECE, NICOLA, ERRA, UGO, and GRUOSSO, MONICA. "Treesketchnet: From sketch to 3d tree parameters generation". *ACM Transactions on Intelligent Systems and Technology* 14.3 (2023), 1–29 1, 4.

[MRWS24] MAHARA, ARPAN, RISHE, NAPHTALI D., WANG, WENJIA, and SADJADI, SEYED MASOUD. "KAN-Attn GAN: Map Generation with Kolmogorov-Arnold Networks and Attention-Based Queries Selection". *2024 International Conference on Machine Learning and Applications (ICMLA)*. 2024, 1599–1604. DOI: 10.1109/ICMLA61862.2024.00247 2.

[MWH*25] MA, XIANPING, WANG, ZIYAO, HU, YIN, et al. "Kolmogorov-Arnold Network for Remote Sensing Image Semantic Segmentation". *arXiv preprint arXiv:2501.07390* (2025) 2.

[NSEL22] NASERENTIN, V., SOMANATH, S., ELEFTHERIOU, O., and LOGG, A. "Combining Open Source and Commercial Tools in Digital Twin for Cities Generation". *IFAC-PapersOnLine* 55.11 (2022). IFAC Workshop on Control for Smart Cities CSC 2022, 185–189. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2022.08.070 2.

[RLM*23] RAISTRICK, ALEXANDER, LIPSON, LAHAV, MA, ZEYU, et al. "Infinite Photorealistic Worlds Using Procedural Generation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, 12630–12641 2.

[SFK*20] SMIRNOV, DMITRIY, FISHER, MATTHEW, KIM, VLADIMIR G., et al. "Deep Parametric Shape Predictions Using Distance Fields". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 2.

[SHD*25] SUN, CHUNYI, HAN, JUNLIN, DENG, WEIJIAN, et al. "3D-GPT: Procedural 3D Modeling with Large Language Models". *2025 International Conference on 3D Vision (3DV)*. 2025, 1253–1263. DOI: 10.1109/3DV66043.2025.00119 2.

[SLK*20] SHARMA, GOPAL, LIU, DIFAN, KALOGERAKIS, EVANGELOS, et al. *ParSeNet: A Parametric Surface Fitting Network for 3D Point Clouds*. 2020. arXiv: 2003.12181 [cs.CV] 2.

[SVS*25] SIMÉONI, ORIANE, VO, HUY V., SEITZER, MAXIMILIAN, et al. *DINOv3*. 2025. arXiv: 2508.10104 [cs.CV]. URL: https://arxiv.org/abs/2508.10104 6.

[TL19] TAN, MINGXING and LE, QUOC. "Efficientnet: Rethinking model scaling for convolutional neural networks". *International conference on machine learning*. PMLR. 2019, 6105–6114 4.

[TRT*22] TCHAPMI, LYNE P., RAY, TRISHIET, TCHAPMI, MICAEL, et al. "Generating Procedural 3D materials from Images using Neural Networks". *Proceedings of the 2022 4th International Conference on Image, Video and Signal Processing*. IVSP '22. Singapore, Singapore: Association for Computing Machinery, 2022, 32–40. ISBN: 9781450387415. DOI: 10.1145/3531232.3531237 2.

[TY25] TANG, WENQIANG and YANG, ZHOUWANG. "UNet-assisted parameterization for B-spline surface approximation". *Computers & Graphics* 132 (2025), 104416. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2025.104416. URL: https://www.sciencedirect.com/science/article/pii/S0097849325002572 2.

[USB22] UNLU, G. E., SAYED, M., and BROSTOW, G. "Interactive Sketching of Mannequin Poses". *Proceedings of the IEEE/CVF International Conference on 3D Vision (3DV)*. Sept. 2022 2.

[WCPM18] WANG, T. Y., CEYLAN, D., POPOVIĆ, J., and MITRA, N. J. "Learning a Shared Shape Space for Multimodal Garment Design". *ACM Trans. Graph.* 37.6 (Dec. 2018). ISSN: 0730-0301. DOI: 10.1145/3272127.3275074. URL: https://doi.org/10.1145/3272127.3275074 2.

[YLY21] YANG, JINGLUN, LI, YOUHUA, and YANG, LU. "Shape transformer nets: Generating viewpoint-invariant 3D shapes from a single image". *Journal of Visual Communication and Image Representation* 81 (2021), 103345. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2021.103345. URL: https://www.sciencedirect.com/science/article/pii/S1047320321002285 2.

[YYW24] YU, RUNPENG, YU, WEIHAO, and WANG, XINCHAO. "Kan or mlp: A fairer comparison". *arXiv preprint arXiv:2407.16674* (2024) 2.

[ZFCW25a] ZHANG, JUSHENG, FAN, YIJIA, CAI, KAITONG, and WANG, KEZE. *Kolmogorov Arnold Fourier Network*. https://github.com/kolmogorovArnoldFourierNetwork/KAF, last accessed on 2025-09-23. 2025 3, 4.

[ZFCW25b] ZHANG, JUSHENG, FAN, YIJIA, CAI, KAITONG, and WANG, KEZE. "Kolmogorov-Arnold Fourier Networks". *arXiv preprint arXiv:2502.06018* (2025) 3.

[ZSL*25] ZHAN, SIMIN, SU, JIAJUN, LIU, PUDU, et al. "Object re-identification using Kolmogorov-Arnold attention networks". *Mathematical Foundations of Computing* (2025) 2.

[ZWH*24] ZHOU, MENGQI, WANG, YUXI, HOU, JUN, et al. "SceneX: Procedural Controllable Large-scale Scene Generation". *arXiv preprint arXiv:2403.15698* (2024) 2.