AC3S: Adaptive Conditioning for 3D-Aware Synthetic Data Generation

Eric Ji[1], Qiran Hu[1], Wufei Ma[2], Sarthak Jain[1], Yingying Li[1], Minh N. Do[1,3], and Yaoyao Liu[1]

[1] University of Illinois Urbana-Champaign
[2] Johns Hopkins University
[3] College of Engineering and Computer Science, VinUniversity

Abstract. Synthetic data generation has emerged as a powerful tool for improving data scalability in computer vision. Recent diffusion-based pipelines have demonstrated strong photorealism. However, how to enforce precise 3D structure and pose consistency in generated images remains challenging. Existing methods leverage visual prompts such as edge maps to guide diffusion models, but often suffer from over-conditioning artifacts that degrade image realism and limit dataset quality. In this paper, we present a diffusion-based image generation framework that enforces 3D structural alignment while preserving photorealism through adaptive conditioning. Our framework, Adaptive Conditioning for 3D-Aware Synthetic Data Generation (AC3S), introduces a self-supervised visual prompt modulator that dynamically adjusts the strength of ControlNet conditioning, preventing over-conditioning and enabling the diffusion model to retain its generative expressiveness. To further enhance diversity and semantic consistency, we develop a multi-agent vision language model framework that composes detailed and 3D-aware prompts aligned with the underlying geometric structure. Together, these components enable the scalable generation of high-quality synthetic datasets with accurate 2D and 3D annotations. Extensive experiments demonstrate that our method significantly improves image quality and downstream utility.[1]

Keywords: Adaptive Conditioning · Diffusion Models · Synthetic Data

1 Introduction

Image generation has rapidly advanced in recent years due to the introduction of diffusion models [4, 5, 10, 16, 29, 34, 40, 44]. Large-scale pre-trained models can synthesize high-fidelity images spanning diverse domains [38,42]. Beyond creative applications, these models have proven to be powerful tools for synthetic data generation, completely transforming the current landscape [14, 33, 35].

Traditional synthetic datasets rely on physics-based rendering engines [7], in which users directly control scene composition, lighting, and camera parameters.

[1] Project page: https://ac3s.cvmlgroup.web.illinois.edu/

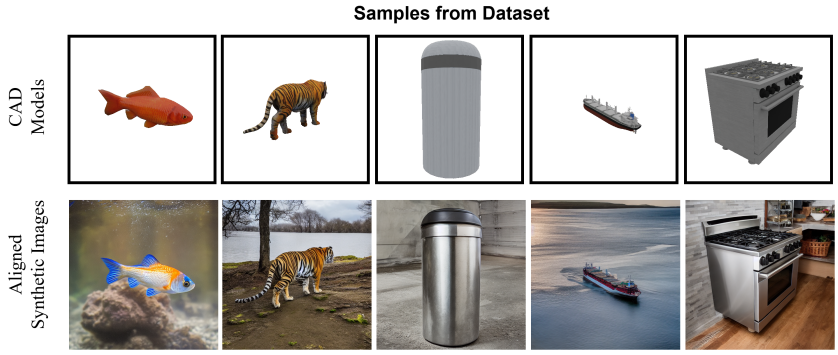


Fig. 1: Illustrated are samples generated using our adaptive pipeline. The top row displays the CAD renders used to extract visual prompts and data labels. We combine the visual prompt with our 3D-aware text prompts to generate the synthetic images in the bottom row. Note that the subject in each synthetic image is precisely aligned with its corresponding render.

Building a photo-realistic scene is achievable, but not a trivial task, especially for large-scale datasets.

Diffusion models alternatively offer highly accessible photo-realistic capabilities through learning an underlying image data manifold. However, popular pre-trained text-to-image models lack reliable and precise 3D control. Recently, 3D-DST [33] bridged this gap with their image generation pipeline. Their method renders an image of a 3D model, extracts its edge map, and treats it as a visual prompt to condition a diffusion model through ControlNet [56]. The objects in their generated images align in pose with the rendered model, allowing them to acquire 3D labels. While accurate poses are achieved, the framework sacrifices photorealism: backgrounds become blurred, object textures are simplified, and scenes become incoherent.

In this paper, we propose a diffusion-based image generation framework that enforces 3D structural alignment while preserving the photorealistic capabilities of pre-trained diffusion models. Our key observation is that visual prompts derived from CAD renders are often sparse and overly restrictive when injected directly into diffusion models. To mitigate this issue, we introduce an adaptive visual prompt modulator that dynamically adjusts the strength of ControlNet conditioning during generation. The modulator predicts a conditioning scale from the feature magnitudes of the diffusion model and ControlNet, enabling the system to automatically balance structural alignment and generative expressiveness.

In addition to visual conditioning, generating diverse and semantically consistent prompts remains an important challenge in large-scale synthetic dataset creation. Template-based prompts often lack detail, while captions extracted from real images may conflict with the visual prompts used for geometric control. To address this issue, we design a multi-agent vision language model framework

that composes detailed and 3D-aware prompts aligned with the underlying visual structure. Multiple specialized agents collaborate to identify the subject, infer pose information, and construct scene descriptions to generate both positive and negative prompts. This design ensures that textual guidance remains consistent with the visual prompts and encourages the diffusion model to generate realistic, diverse images.

Qualitatively, our pipeline's adaptive conditioning modules yield superior samples compared to the baseline. We validate such observations through a series of experiments analyzing generative quality and downstream utility. Specifically, we observe an improvement in FID score [15] by 15.95 points over the baseline and provide up to a 7% boost in accuracy when fine-tuning models with our synthetic data. Furthermore, we show that in the absence of real-world data, training strictly on our synthetic data can still produce functional models.

Our contributions can be summarized as follows:

- **Visual prompt modulator for ControlNet.** We introduce a simple yet effective self-supervised training strategy to train a visual prompt modulator. By automatically attenuating ControlNet outputs, we can prevent overconditioning on a per-sample basis, leading to higher quality images.
- **Multi-agent VLM prompt generator.** We propose a novel multi-agent framework to synthesize descriptive and diverse text prompts. By integrating 3D context, we can ensure alignment between text and visual prompts.
- **Full Synthetic Dataset.** Our full synthetic dataset consists of one million images evenly distributed among the 1,000 ImageNet [9] categories. We provide labels for classification, object pose estimation, and additional metadata.

2 Related Work

Text-to-Image Diffusion Models. Pre-trained text-to-image diffusion models excel in generating photo-realistic images from natural language [1, 17, 38, 42, 43, 46, 47]. These models begin with sampling Gaussian noise, before iteratively passing through a denoising U-Net [41] until reaching the image manifold. Latent diffusion models [40] further optimize by operating in a compressed latent representation, addressing the high-dimensionality of images. Conditioning on text allows for guidance during generation towards desired results. Text is highly expressive, but it typically cannot provide fine-grained constraints. As a result, standard text-to-image models struggle with accurate geometric control, motivating alternative prompting techniques like ControlNet [56].

ControlNet. ControlNet [56] extends the conditioning capabilities of diffusion models with visual prompts. Examples of visual prompts include edge maps, depth maps, and segmentation masks. Their method is inspired by side-tuning [54], an incremental learning [10,11,19–21,23–28,32,57,60] technique that allows networks to adapt to new tasks by training a lightweight "side" network.

For diffusion models, a copy of the U-Net encoder [41] acts as the "side" network. This ControlNet takes the visual prompt as input and learns to produce features to inject back into the decoder through skip connections. The injections are performed by element-wise addition

$$\hat{f} = f + \mathcal{F}(I_{prompt}; \theta) \tag{1}$$

where f denotes the intermediate U-Net features, and $\mathcal{F}(v; \theta)$ denotes the ControlNet outputs produced by visual prompt v with parameters θ. When training a ControlNet, the diffusion model's weights are frozen, allowing only the parameters θ to learn. In practice, we observe that features produced by pre-trained ControlNets are too strong and degrade photorealism, hence motivating our adaptive modulator.

Vision Language Models (VLMs). Vision language models (VLMs) [2, 36, 45, 55, 58–60] have experienced widespread adoption with the introduction of GPT-4 [36] and QWEN [45]. Their ability to interpret both natural language and images makes them shine in descriptions and complex reasoning. VLMs generate their output by autoregressively generating one word at a time. Compared to prompting a single VLM, a multi-agent framework can decompose and more efficiently coordinate complex tasks. We assemble a team of agents to construct 3D-aware text prompts for image generation.

3D-DST. 3D-DST [33] introduces a general framework for producing synthetic datasets with geometric control. Their method renders images of CAD models from diverse viewpoints to extract edge maps, which are used as visual prompts for ControlNet. Combined with text captions, their pipeline enables the generation of synthetic images with 3D annotations. While effective at enforcing geometric consistency, the lack of flexibility in their prompting leads to visible artifacts, the core problem our adaptive conditioning modules solve.

3 Method

Our method adaptively controls both visual and textual conditioning during image generation. By automatically tuning conditioning signals, our approach brings scalable solutions to limitations associated with prior pipelines. In particular, our proposed pipeline mitigates visual overconditioning, which manifests as low textures, and ensures consistency between visual and textual guidance. We begin by introducing the procedure for collecting visual prompts on a large-scale. Then, we discuss the motivation for the visual prompt modulator and the self-supervised training algorithm. Finally, we will lay out the design of our multi-agent VLM text prompt generator.

3.1 Acquiring Visual Prompts Through Rendering

Following 3D-DST [33], we obtain visual prompts through graphics-based rendering. All 3D models are sourced from large-scale CAD repositories, including

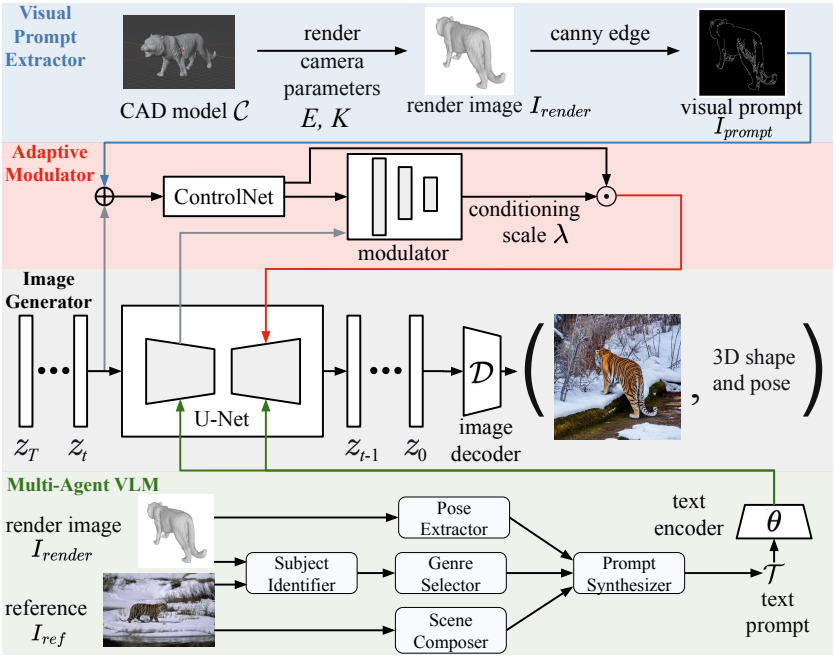


Fig. 2: Our full generation pipeline is composed of four interconnected modules. (1) Visual Prompt Extractor: For a given 3D CAD model, we sample a set of camera parameters E, K to render a 2D image I_{render}. The visual prompt I_{prompt} is extracted from I_{render} in the form of an edge map and serves as the input to ControlNet. (2) Adaptive Modulator: Instead of directly injecting ControlNet's output into the diffusion model, we treat it as input to our modulator alongside diffusion model features to predict the conditioning scale λ. Once attenuated by λ, ControlNet outputs are injected. (3) Multi-Agent VLM: The render I_{render} and real image reference I_{ref} are simultaneously processed by multiple agents to extract pose, color, and scene information for composing text prompt $\mathcal{T}$. (4) Image Generator: Our semantically consistent prompts are used to guide the generation of the photorealistic image.

ShapeNet [6], Objaverse [8], and OmniObject3D [50], resulting in approximately 30 CAD models for each class category. For each model C, we randomly sample a camera viewpoint to render image I_{render} using an off-the-shelf renderer R [7].

$$I_{render} = R(C, E, K) \tag{2}$$

The camera's intrinsics K are based on a perspective camera with a focal length of 35 mm. Camera extrinsics E are sampled through varying distance, azimuth, and elevation. The distribution of viewpoints has defined constraints to avoid implausible perspectives that would not appear in real-world imagery. I_{render} is not intended to be photorealistic, requiring no modifications to materials, lighting, or background. Instead, it is used as a source of geometric structure. We use a Canny edge detector [3] to extract an edge map for the visual prompt,

$$I_{prompt} = CannyEdge(I_{render}) \tag{3}$$

We explored alternative visual prompts, such as depth, but found that edge maps consistently yielded the highest quality samples.

3.2 Adaptive Visual Prompt Modulator

Most visual prompts are sparse in nature, particularly when computed from material-less and background-less renders. Feeding these prompts into ControlNet [56] leads to overconditioning, manifesting as oversimplified textures and uninteresting scenes. Injecting unmodulated ControlNet outputs into our diffusion model heavily restricts the pre-trained generative prior that we want to leverage.

In our study, we find that adding textual conditioning helps lower a diffusion model's denoising objective. Interestingly, this contrasts with injecting visual conditioning, which hurts the loss. This suggests ControlNet introduces harmful noise to the system. We find that attenuating ControlNet outputs can directly mitigate the damage. This appears as increased diversity of backgrounds, richer details, and an overall greater realism.

We define this attenuation using a conditioning scale λ where f denotes the intermediate U-Net features of the diffusion model, and $\mathcal{F}(I_{prompt}; \theta)$ denotes the ControlNet outputs conditioned on visual prompt I_{prompt}.

$$\hat{f} = f + \lambda \cdot \mathcal{F}(I_{prompt}; \theta) \tag{4}$$

Manually tuning λ can be effective for small batches, but scaling up for dataset generation would be infeasible. We find that the optimal λ cannot generalize across class categories and even varies with different random seeds for the same I_{prompt}. However, we do note a correlation between λ, f, and $\mathcal{F}(I_{prompt}; \theta)$. This motivated us to introduce our adaptive visual prompt modulator that predicts λ dynamically based on $\|f\|$ and $\|\mathcal{F}(v; \theta)\|$ at the first timestep. Our implementation for the modulator is as a lightweight three-layer Multi-Layer Perceptron (MLP) as seen in Fig. 2.

Despite similar modulators [22,52] employing a denoising loss for supervision, we have shown it is problematic as the objective is trivially minimized with $\lambda = 0$. To avoid a degenerate solution, we use self-supervision with pseudo-labels. Since λ is continuous, we discretize into candidate bins. λ values in the range $[0.3, 1.0]$ with increments of 0.1 are considered. We find that images generated with values below 0.3 almost never align with the visual prompt. When generating a sequence of images with λ spanning the range, we observe that the desired 3D structure is suddenly enforced once λ exceeds a threshold. Images in the sequence with λ below this threshold behave as if barely affected by ControlNet. While pose exhibits a sudden transition, image quality gradually decreases as λ increases. This phenomenon allows us to define the optimal λ as the smallest λ above the threshold.

Algorithm 1 Optimal Visual Conditioning Scale - Pseudo-Labeling

Input: object renders $\{r_i\}_{i=1}^{n}$, candidate scales $\Lambda = \{0.30, 0.40, 0.50, \ldots, 1.0\}$
for each render r_i **do**
  Extract edge map v_i from r_i as visual prompt
  for each $\lambda \in \Lambda$ **do**
    Generate image I_λ using ControlNet with scale λ and prompt v_i
  end for
  Cluster $\{I_\lambda : \lambda \in \Lambda\}$ using k-means ($k = 2$)
  Let $C_{\lambda=1}$ be the cluster label of $I_{\lambda=1}$
  Optimal $\lambda^* = \min_{\lambda \in \Lambda}$ s.t. cluster$(I_\lambda) = C_{\lambda=1}$
end for

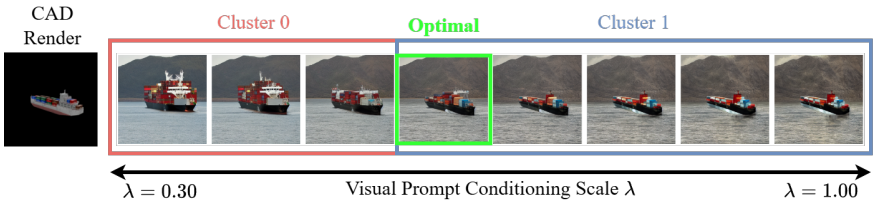


Fig. 3: Performing clustering across a sequence of images generated with varying conditioning scales can be used to identify the optimal conditioning scale.

We similarly define the optimal λ as the point where the change in pose becomes distinguishable, which is analogous to the concept of "just noticeable difference" (JND). When provided with two images, JND refers to when a visible attribute changes just enough such that it is distinguishable by human observers [53]. In our setting, the relevant attribute is pose, but other attributes can include shapes, expressions, or material properties.

We utilize Algorithm 1 to acquire these pseudo-labels. For a render, we obtain the visual and text prompts to generate a sequence of images across a sweep of λ values. Images in the sequence naturally observe two distinct distributions: (i) poorly aligned images at low λ, and (ii) aligned images with λ above a threshold. This allows us to apply k-means clustering with $k = 2$ on the pixel-level features to detect the JND. After clustering, we identify the cluster containing the image produced with $\lambda = 1.0$ and designate it as the aligned cluster. The optimal λ, is then selected as the smallest λ value within that cluster. Fig. 3 provides a visualization of this algorithm. We repeat this procedure across all class categories to obtain a set of pseudo labels, which are used to train the modulator without the need for any human annotations.

3.3 Multi-Agent VLM Text Prompt Generator

We introduce a multi-agent VLM system designed to synthesize text prompts that are semantically consistent with our visual prompts. This framework addresses several limitations associated with current approaches. Early methods [14]

rely on fixed templates (e.g., "A high-quality image of a [CLASS]"), often lacking granular details and providing insufficient guidance for synthesizing complex scenes. More recent works mitigate this by leveraging VLMs as captioners to extract descriptions from large-scale image datasets [33, 35]. While these captions are rich in detail, they frequently are not consistent with our visual prompts. For example, a sampled caption may describe a close-up scene containing multiple object instances, whereas our visual prompt depicts a holistic view of a singular instance. These conflicting text and visual prompts can produce degrading hallucinations. We orchestrate a team of specialized agents to analyze inputs, extract information, and compose quality prompts.

In our multi-agent system, each VLM acts as an independent expert defined by its role and domain expertise. Agents are profiled with detailed specifications, including input modalities, output schema, and behavioral constraints. We will highlight each agent and their objectives:

Agent 1 (Subject Identifier) takes the render I_{render} and a real reference image I_{ref} from ImageNet [9] to identify a subject label (1-3 words). Relying purely on basic ImageNet class categories instead of detailed labels can leave out meaningful context for downstream agents. By having access to both images, Agent 1 maximizes classification capabilities through cross-referencing before committing to a semantic anchor.

Agent 2 (Genre Selector) selects a photography genre from a pre-defined set (e.g., wildlife, macro, etc.) that would best align with the identified subject. Photography genres encode a prior on style and other conventions.

Agent 3 (Pose Extractor) takes the render I_{render} and extracts the subject and camera pose. Since the edge map is extracted from the same render, using this information for our prompt will ensure alignment in perspective.

Agent 4 (Scene Composer) takes the reference image I_{ref} to extract visual attributes for the subject (color, texture, etc.) and the background scenery. Having information from the reference allows prompts to capture details and diversity present in ImageNet.

Agent 5 (Positive Prompt Synthesizer) assembles the extracted information from agents 2-4 into a coherent positive prompt.

Agent 6 (Protected Terms Identifier) identifies keywords and phrases to exclude from the negative prompt. Synonyms from our positive prompt should be protected and avoided in the negative prompts to prevent inadvertently guiding the generation process away from the desired result.

Agent 7 (Negative Prompt Synthesizer) assembles a negative prompt by identifying common negative terms applicable to our subject, while ensuring they do not include protected terms produced by Agent 6.

3.4 Prompt Engineering for Agents

We build each agent using well-established prompt-engineering principles like chain-of-thought prompting [48] and structured output generation [49]. Each agent operates according to a structured prompt template consisting of seven

Table 1: To ensure each agent specializes in their designated tasks, we provide structured prompts following the format outlined in this table.

Section	Purpose
Role	Initializes an expert persona aligned with the agent's specialized task.
Goal	Provides a single-sentence objective that defines success criteria.
Context	Describes input modalities and their semantic content.
Constraints	Enumerates explicit behavioral boundaries to eliminate uncertainty.
Examples	Provides paired positive and negative demonstrations with rationales.
Format	Specifies the expected structure of outputs for reliable parsing.
Criteria	Defines acceptance criteria for self-verification.

standardized sections: Role, Goal, Context, Constraints, Contrastive Examples, Output Format, and Criteria [39]. Table 1 outlines the purpose of each section.

The multi-agent workflow is implemented using LangGraph's StateGraph abstractions, which provide directed acyclic graph execution with type state management. We employ Qwen3-VL-23B-Instruct [45] with 4-bit quantization. Agents communicate through a shared typed state dictionary following the shared message pool pattern introduced by MetaGPT [18]. The communication structure combines sequential and parallel workflows as certain agents require dependencies from upstream agents. The workflow for producing the positive prompt is visualized in Fig. 2.

3.5 Dataset Filtering

Filtering out poorly generated samples from a dataset significantly boosts performance for downstream tasks. 3D-DST [33] trains a pose estimator with k-fold cross-validation to assign a loss-based score for each image. For smaller datasets, it may be difficult for the pose estimator to converge and produce a meaningful score. It is also extremely computationally expensive for large datasets, as it requires a designated model for each class category. We instead leverage DreamSim [12], an ensemble of foundation models fine-tuned to capture a holistic perceptual metric. Unlike traditional metrics, which target low-level features, DreamSim score primarily captures object pose and semantic information. To evaluate generated samples, we compute the perceptual similarity between each synthetic image and its corresponding render image I_{render}. We ensure the metric remains invariant to color by applying a grayscale transformation to both images before scoring. We find that using the DreamSim score as a ranking mechanism can efficiently allow us to identify the strongest samples for our dataset.

4 Experiments

To evaluate the effectiveness of our adaptive prompting techniques, we generate a dataset using the proposed pipeline and conduct several comprehensive experiments. We first conduct an ablation study analyzing the impact of individual

modules on image quality. Beyond a qualitative assessment, we validate the utility of our dataset through studying its impact on both 2D and 3D computer vision tasks. We benchmark our methods against multiple existing synthetic datasets.

4.1 Image Quality Ablation

We conduct an ablation study for each module of our method to demonstrate the impact on image quality and diversity. We quantify quality and diversity through the Frechet Inception Distance (FID) score [15], which measures the distance between the distribution of our generated images and real images. In this experiment, we compare 1,000 synthetic images from each method to 1,000 images from ImageNet [9] for 50 classes. The results are outlined in Table 2. The baseline method is 3D-DST [33], to which we incorporate the adaptive visual prompt modulator and multi-agent VLM text prompt generator to produce our method. When generating the data across methods, we manually fixed the seeds to ensure fair comparisons. Analyzing the results, we can observe that both the visual prompt modulator and text prompt generator can improve the overall FID score by 9.66 and 8.26 points, respectively. When working in conjunction for our full pipeline, we observe the greatest overall improvement of 15.95 points. We can also note that our method particularly boosts image quality and diversity for animal classes with a 23.8 point improvement. Figure 4 illustrates example images from this ablation study.

4.2 Synthetic Datasets

In our model training experiments, we use AC3S-50, a 50-class dataset generated with our adaptive conditioning modules. The distribution of classes contains a mixture of animals and objects following the ImageNet [9] distribution. For our diffusion model, we use Stable Diffusion v1.5 and sample 20-timestep trajectories. We benchmark this dataset against two existing synthetic datasets, Text2Img [33], which we generate following their implementation and codebase. For each synthetic dataset, we have 1,000 images per class, equating to a total of 50,000 images each.

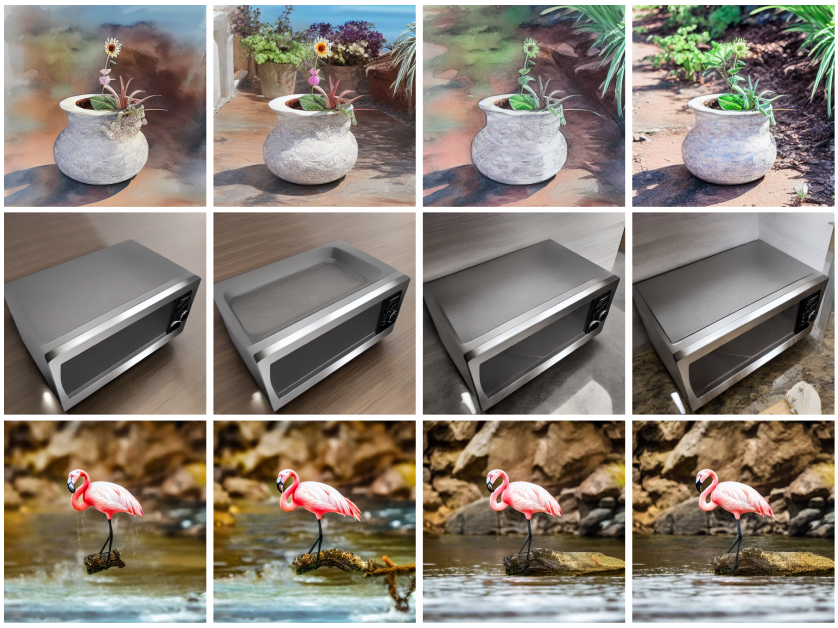
4.3 Synthetic Pre-training for Classification Implementation

We evaluate the utility of synthetic data for classification across three methods. The baseline method is trained using only our real data from ImageNet [9]. The remaining three methods incorporate synthetic data during pre-training.

The most common case of synthetic data is to pre-train a model before fine-tuning it on the target domain with real data. In our classification experiments, we use ResNet-18 [13], Swin-S [30], and ConvNeXt-S [31] to evaluate our dataset across convolutional and transformer-based backbones. All models are trained completely from scratch with default PyTorch [37] weight initialization.

Table 2: Results from our ablation study analyzing the impact of each adaptive conditioning module.

Method	Modulator	VLM Prompts	FID Score (↓)		
			Animals	Objects	Overall
Baseline (3D-DST [33])			72.67	90.38	87.19
+ Visual Prompt Modulator	✓		60.93	81.18	77.53
+ Text Prompt Generator		✓	54.68	84.23	78.93
Ours	✓	✓	**48.87**	**76.15**	**71.24**



(1) Baseline (3D-DST) (2) + Visual Prompt Modulator (3) + Text Prompt Generator **(4) Ours**

Fig. 4: Visualizations of various classes from the ablation study. For the flamingo in the last row, our method (4) uses both the visual prompt modulator and text prompt generator simultaneously to produce the most realistic and detailed image. Compared to (3), introducing the visual prompt modulator in (4) brings out greater texture and details in both the bird itself and the rocky background. Simultaneously, using the prompts from our text prompt generator in (3) results in more realistic colors and plausible scenery compared to (1).

For ResNet-18, we fine-tune for 50 epochs on each synthetic dataset, which was sufficient for model convergence. We select the best pre-trained weights from the epoch with the lowest validation loss. We follow the same pre-training procedure for Swin-S and ConvNeXt-S, which required 150 epochs and 200 epochs, respectively. Following pre-training, each model is fine-tuned on ImageNet [9] until

Table 3: Reported top-1 classification accuracies on ImageNet [9]. Each method, except for Baseline, was pre-trained on synthetic data before being fine-tuned to ImageNet.

	Animals			Objects			Overall		
Method	ResNet	ConvNeXt	Swin	ResNet	ConvNeXt	Swin	ResNet	ConvNeXt	Swin
Baseline	91.30	94.03	96.16	73.08	77.05	78.37	76.34	80.12	81.59
Text2Img [14]	91.89	93.52	96.33	73.80	76.05	78.05	77.08	79.21	81.36
3D-DST [33]	91.98	**95.39**	96.25	73.63	78.90	80.28	76.96	81.89	83.17
AC3S(Ours)	**92.66**	95.14	**96.76**	**73.93**	**80.01**	**81.41**	**77.33**	**82.75**	**84.19**

convergence, with the final model chosen from the epoch with the lowest validation loss. ResNet-18 and Swin-S each required 150 epochs, while ConvNeXt-S needed 500 epochs to converge.

4.4 Synthetic Pre-training for Classification Results

We report all results as top-1 accuracies evaluated on a held-out test split of ImageNet [9]. Table 3 summarizes the results for pre-training on synthetic data before fine-tuning with real data. We observe that across all three architectures, using our data for pre-training yields the greatest overall improvement. ConvNeXt-S [31] and Swin-S [30] both saw a significant increase of 2.6%, and ResNet-18 [13] gained 0.99%.

4.5 Synthetic Pre-training for Pose Estimation Implementation

We perform a similar pre-training procedure for object pose estimation to validate the accuracy of our 3D annotations. Instead of ImageNet [9], we fine-tune and evaluate with PASCAL3D+ [51], which consists of approximately 30,000 images spread across 10 class categories. We selected 3 class categories that intersected with our AC3S-50 dataset: sofa, dining table, and car. For each class category and method, we trained a designated pose estimator with ResNet-18 [13] backbone. Pre-training required 20 epochs for model convergence, and fine-tuning to PASCAL3D+ required 10.

4.6 Synthetic Pre-training for Pose Estimation Experiment Results

We report all results as threshold accuracies in Table 4 based on the geodesic distance between the predicted and ground-truth rotation matrix. Across all three class categories, our method consistently yielded greater improvement over 3D-DST [33]. On average, we observe a 1.23% gain for $\pi/6$ threshold and 2.17% gain for $\pi/18$. We find the biggest improvement over the baseline to be sofa for $\pi/18$ with a jump of 7.39% by pre-training with AC3S. In Table 5, without any fine-tuning, the gap performance between 3D-DST and AC3S is immense, implying that AC3S better resembles the distribution of PASCAL3D+.

Table 4: Reported accuracies measure the geodesic distance between predicted and ground-truth rotation matrices with a $\pi/6$ and $\pi/18$ error threshold on PASCAL3D+ [51]. Each method, except for Baseline, was pre-trained on synthetic data before being fine-tuned to PASCAL3D+.

	Sofa		Dining Table		Car	
Method	Acc@$\pi/6$	Acc@$\pi/18$	Acc@$\pi/6$	Acc@$\pi/18$	Acc@$\pi/6$	Acc@$\pi/18$
ResNet (Baseline)	80.79	34.98	79.86	49.83	87.60	66.19
w/ 3D-DST [33]	86.70	37.83	80.20	51.88	88.51	67.62
w/ AC3S (Ours)	**88.18**	**38.92**	**81.91**	**53.92**	**89.03**	**71.02**

Table 5: Reported accuracies measure the geodesic distance between predicted and ground-truth rotation matrices with a $\pi/6$ and $\pi/18$ error threshold on PASCAL3D+ [51]. Each method was only trained on synthetic data.

	Sofa		Dining Table		Car	
Method	Acc@$\pi/6$	Acc@$\pi/18$	Acc@$\pi/6$	Acc@$\pi/18$	Acc@$\pi/6$	Acc@$\pi/18$
w/ 3D-DST [33]	10.84	0.99	14.68	2.05	7.05	0.39
w/ AC3S (Ours)	**34.48**	**10.34**	**33.79**	**5.46**	**47.91**	**17.49**

5 Conclusion

In this paper, we present AC3S, a 3D-aware image generation pipeline centered on adaptive conditioning. Our self-supervised visual prompt modulator adaptively attenuates ControlNet [56] outputs, effectively preventing over-conditioning. This allows us to strike a balance between geometric alignment and generative expressiveness. Simultaneously, our multi-agent VLM system establishes a novel approach to producing text prompts that remain consistent with our visual prompt. Experimentation confirms that when these two adaptive conditioning modules are integrated together, we not only get a significant boost in image quality, but also increased utility for downstream 2D and 3D vision tasks. We believe AC3S provides many meaningful insights into scaling and advancing synthetic image generation.

Acknowledgements. This research is supported by the National Artificial Intelligence Research Resource Pilot Awards NAIRR250199, NAIRR260019, and NAIRR260077, the AMD University Program's AI & HPC Cluster, NVIDIA Academic Grant Program, and Lambda's Research Grant. It is also supported in part by the Center for Networked Intelligent Components and Environments, a partnership between the University of Illinois Urbana-Champaign and Foxconn Interconnect Technology. Computational resources are also provided by Delta and DeltaAI at the National Center for Supercomputing Applications through ACCESS allocations CIS250012, CIS250816, and CIS251188.

References

1. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. arXiv preprint arXiv:2304.08466 (2023)
2. Bordes, F., Pang, R.Y., Ajay, A., Li, A.C., Bardes, A., Petryk, S., Mañas, O., Lin, Z., Mahmoud, A., Jayaraman, B., et al.: An introduction to vision-language modeling. arXiv preprint arXiv:2405.17247 (2024)
3. Canny, J.: A computational approach to edge detection. TPAMI pp. 679–698 (2009)
4. Cao, W., Zhang, H., Tang, J., Wu, Y., Li, Y., Yu, N., Wang, S., Liu, Y.: Redirect4d-bench: A scalable benchmark for camera redirection of monocular dynamic videos with pseudo-4d ground truth (2026)
5. Cao, W., Zhang, H., Tian, F., Wu, Y., Li, Y., Wang, S., Yu, N., Liu, Y.: FreeOrbit4D: Training-free arbitrary camera redirection for monocular videos via foreground-complete 4D reconstruction. In: SIGGRAPH (2026)
6. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
7. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018)
8. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR. pp. 13142–13153 (2023)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
10. Duan, R., Chen, J., Kortylewski, A., Yuille, A., Liu, Y.: Prompt-based exemplar super-compression and regeneration for class-incremental learning. In: BMVC (2025)
11. Fischer, T., Liu, Y., Jesslen, A., Ahmed, N., Kaushik, P., Wang, A., Yuille, A.L., Kortylewski, A., Ilg, E.: inemo: Incremental neural mesh models for robust class-incremental learning. In: ECCV. pp. 357–374 (2024)
12. Fu, S., Tamir, N., Sundaram, S., Chai, L., Zhang, R., Dekel, T., Isola, P.: Dreamsim: Learning new dimensions of human visual similarity using synthetic data. arXiv preprint arXiv:2306.09344 (2023)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
14. He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., Qi, X.: Is synthetic data from generative models ready for image recognition? arXiv preprint arXiv:2210.07574 (2022)
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017)
16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS pp. 6840–6851 (2020)
17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) NeurIPS (2020)
18. Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., Zhou, L., Ran, C., Xiao, L., Wu, C., Schmidhuber, J.: Metagpt: Meta programming for a multi-agent collaborative framework. In: The twelfth ICLR (2023)

19. Li, Y., Chen, X., Li, N.: Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. Advances in Neural Information Processing Systems **32** (2019)
20. Li, Y., Das, S., Li, N.: Online optimal control with affine constraints. In: AAAI. pp. 8527–8537 (2021)
21. Li, Y., Qu, G., Li, N.: Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. IEEE Transactions on Automatic Control **66**(10), 4761–4768 (2020)
22. Lin, Q., Sun, X., Gao, Y., Zhong, Y., Zhao, Z., Li, D., Wang, H.: Tasr: Timestep-aware diffusion model for image super-resolution. In: ACM Multimedia. pp. 10034–10043 (2025)
23. Liu, Y., Li, Y., Schiele, B., Sun, Q.: Online hyperparameter optimization for class-incremental learning. In: AAAI. pp. 8906–8913 (2023)
24. Liu, Y., Li, Y., Schiele, B., Sun, Q.: Wakening past concepts without past data: Class-incremental learning from online placebos. In: WACV. pp. 2226–2235 (2024)
25. Liu, Y., Schiele, B., Sun, Q.: Adaptive aggregation networks for class-incremental learning. In: CVPR. pp. 2544–2553 (2021)
26. Liu, Y., Schiele, B., Sun, Q.: Rmm: Reinforced memory management for class-incremental learning. NeurIPS **34**, 3478–3490 (2021)
27. Liu, Y., Schiele, B., Vedaldi, A., Rupprecht, C.: Continual detection transformer for incremental object detection. In: CVPR. pp. 23799–23808 (2023)
28. Liu, Y., Su, Y., Liu, A.A., Schiele, B., Sun, Q.: Mnemonics training: Multi-class incremental learning without forgetting. In: CVPR. pp. 12245–12254 (2020)
29. Liu, Y., Sun, Q., He, X., Liu, A., Su, Y., Chua, T.: Generating face images with attributes for free. TNNLS **32**(6), 2733–2743 (2021)
30. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
31. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR. pp. 11976–11986 (2022)
32. Luo, Z., Liu, Y., Schiele, B., Sun, Q.: Class-incremental exemplar compression for class-incremental learning. In: CVPR. pp. 11371–11380 (2023)
33. Ma, W., Liu, Q., Wang, J., Wang, A., Yuan, X., Zhang, Y., Xiao, Z., Zhang, G., Lu, B., Duan, R., Qi, Y., Kortylewski, A., Liu, Y., Yuille, A.: Generating images with 3d annotations using diffusion models. In: ICLR (2023)
34. Ma, W., Zhang, G., Liu, Q., Zeng, G., Kortylewski, A., Liu, Y., Yuille, A.: Imagenet3d: Towards general-purpose object-level 3d understanding. In: NeurIPS (2024)
35. Nguyen, Q., Vu, T., Tran, A., Nguyen, K.: Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. NeurIPS pp. 76872–76892 (2023)
36. OpenAI: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
37. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NeurIPS 2017 Autodiff Workshop (2017)
38. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
39. Robino, G.: Conversation routines: A prompt engineering framework for task-oriented dialog systems. arXiv preprint arXiv:2501.11613 (2025)

40. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
41. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241 (2015)
42. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, S.K.S., Lopes, R.G., Ayan, B.K., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. NeurIPS pp. 36479–36494 (2022)
43. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., Parikh, D., Gupta, S., Taigman, Y.: Make-a-video: Text-to-video generation without text-video data. In: ICLR (2023)
44. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
45. Team, Q.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)
46. Trabucco, B., Doherty, K., Gurinas, M., Salakhutdinov, R.: Effective data augmentation with diffusion models. arXiv preprint arXiv:2302.07944 (2023)
47. Villegas, R., Babaeizadeh, M., Kindermans, P., Moraldo, H., Zhang, H., Saffar, M.T., Castro, S., Kunze, J., Erhan, D.: Phenaki: Variable length video generation from open domain textual descriptions. In: ICLR (2023)
48. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. NeurIPS pp. 24824–24837 (2022)
49. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
50. Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., Lin, D., Liu, Z.: Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In: CVPR. pp. 803–814 (2023)
51. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: WACV. pp. 75–82. IEEE (2014)
52. Yehezkel, S., Dahary, O., Voynov, A., Cohen-Or, D.: Navigating with annealing guidance scale in diffusion space. In: SIGGRAPH Asia. pp. 1–11 (2025)
53. Yu, A., Grauman, K.: Just noticeable differences in visual attributes. In: ICCV. pp. 2416–2424 (2015)
54. Zhang, J.O., Sax, A., Zamir, A., Guibas, L., Malik, J.: Side-tuning: a baseline for network adaptation via additive side networks. In: ECCV. pp. 698–714. Springer (2020)
55. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: A survey. TPAMI **46**(8), 5625–5644 (2024)
56. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV. pp. 3836–3847 (2023)
57. Zhang, Y., Li, X., Chen, H., Yuille, A.L., Liu, Y., Zhou, Z.: Continual learning for abdominal multi-organ and tumor segmentation. In: MICCAI. pp. 35–45 (2023)
58. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. IJCV **130**(9), 2337–2348 (2022)
59. Zhu, D., Shen, X., Li, X., Elhoseiny, M., et al.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. In: ICLR. vol. 2024, pp. 18378–18394 (2024)
60. Zhu, Z., Gong, Y., Xiao, Y., Liu, Y., Hoiem, D.: How to teach large multimodal models new skills? In: ECCV (2026)