Subject Specific Fine-tuning to Create Synthetic Datasets for Computer Vision in Sports

Arnav Gangal (agangal)

1 Introduction and Motivation

All code for this project can be found at `https://github.com/arnavgangal/baseball-project`.

1.1 Motivation

Human pose estimation is a task in computer vision, designed to track the limbs and joints of humans in still images and videos. It is a problem with a highly active body of research, and its applications include human-computer interaction and realistic augmented reality [Zheng et al., 2023]. Advances in neural network architecture designs and improved access to computational resources over the last five years have led to major leaps in model accuracy and applicability. However, existing research is often limited by a lack of sufficiently diverse data [Purkrábek and Matas, 2023]. In particular, existing human pose estimation implementations often rely on datasets that contain only images from front-on viewpoints, and require manual annotation in order to be valuable as training data. When confronted with novel viewpoints, particularly viewpoints in which all or part of the subject's body is occluded, models trained on typically available data often struggle [Sárándi et al., 2018].

To overcome the scarcity of sufficiently diverse training data, it may be possible to use deep generative models such as stable diffusion to supplement existing training datasets. These supplemented datasets could then be used in sports like baseball to train models that are robust to unexpected obstructions, backgrounds, or viewpoints. In practice, such models have been used for early detection and prediction of injuries, potentially saving professional sports teams millions of dollars in lost revenue and replacement players [Piergiovanni and Ryoo, 2019]. These models are also highly costly to train, and the conventional data required to train them is expensive to collect and access. The use of synthetic data to supplement training human pose estimation models has already seen considerable success, and an example of a model trained on purely real data versus one trained on a combination of real and synthetic data can be seen in figure 1. However, there is no academic consensus on the most cost and time-effective way to generate these datasets.

Overall approach To address this data gap, I fine-tuned Stable Diffusion on a dataset of baseball pitching images, using LoRA (Low-Rank Adaptation) to introduce the pose of throwing a pitch into the Stable Diffusion model. To achieve this, I collected around 100 images of baseball players throwing pitches from MLB's official photo stream, from a variety of angles. My main focus with collecting these images was to get a diversity of pitching poses and angles so that different parts of the body would be occluded.

To further refine the fine-tuning process, I also use textual inversion to add a 'pseudo-word' to the model's vocabulary, with the objective of associating that word with the concept of pitching. Evaluation of these models was done both qualitatively and quantitatively. Qualitatively, images generated using the baseline stable diffusion checkpoint were compared to images generated from models with either textual inversion or LoRA weights added. Quantitatively, I used the CLIP score to assess how compatible generated images from each type of model were with the captions used to generate them, and use the Frechet Inception Distance (FID) to compare the models with the real image dataset on which they were trained.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

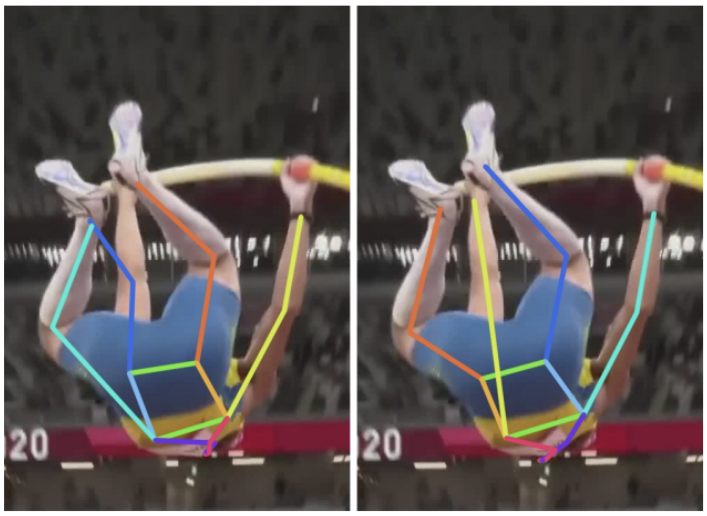

Figure 1: A model trained on a real dataset on the right, compared to a model trained on a combination dataset on the left, [Purkrábek and Matas, 2023]

Both of the fine-tuning methods that I investigated have some desirable characteristics - LoRA checkpoints do not store entire trained models, but instead apply small changes to checkpoints. This results in much smaller file sizes, making them easy to distribute and incorporate into already existing models and pipelines. Additionally, textual inversions do not require large datasets - empirically, they have been shown to learn faces, styles, and objects extremely well [Gal et al., 2022] from between 3 and 5 images.

It is important to note that using stable diffusion as a data-generating tool can only address one part of the data gap. These images still need to be labeled, and in practice, this has been done using a combination of manual annotation and unsupervised learning [Chen et al., 2023].

1.2 Related work

Using synthetic data as training input for domain-specific machine learning tasks is an active area of research, with constant improvements to domain-specific data generation pipelines. Synthetic data can overcome challenges of data scarcity, data of poor quality, and issues of data privacy. As a result, models trained on purely real data can suffer from underfitting [Lu et al., 2023]. Empirical results suggest that by integrating synthetic data into the training process, model robustness can be bolstered, resulting in a more comprehensive model that transcends the limitations of insufficient or inadequate data. In fields such as high energy particle physics, the ATLAS experiment at CERN has found success in using both VAEs and GANs to simulate the responses of one of the particle colliders calorimeters to particle collisions, and found that both of these models were able to simulate collisions with accurate total energies [Collaboration, 2022]. Calculating the energies of particle collisions using traditional simulators is highly expensive, and the effectiveness of deep generative models in simulating collisions may provide a way to reduce these costs.

The concept of synthetic data generated using specifically diffusion models is also not novel - particularly in medical imaging cases, fine-tuned diffusion models (using LoRA) have been found to be effective tools for generating synthetic MRI and CT Scan data [Gu et al., 2023]. Diffusion methods have also been used for anomaly detection in brain imaging [Behrendt et al., 2023], by inpainting erased patches of training data. These models have been observed to improve upon anomaly detection benchmarks set by VAEs and GANs by up to 25%, suggesting that they are able to learn sufficiently detailed representations of subject-specific data.

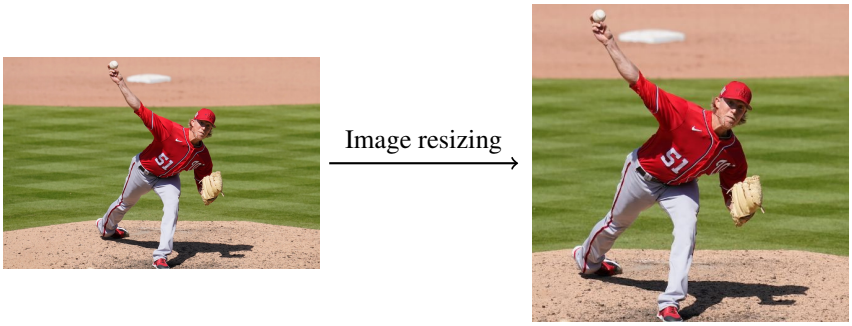

Figure 2: Image processing workflow

2 Problem statement

In this problem, we are using MLB photo stream data of baseball pitchers to finetune a stable diffusion model, using both LoRA and textual inversion. The goal of the project is to finetune a model that can produce realistic looking baseball pitching actions, to be used in robust computer vision models and human pose estimation models as synthetic training data.

All of the fine-tuning methods will be trained on top of a base stable diffusion model. In particular, I will be using the stable-diffusion 1.5 checkpoint (available here), and using the diffusers package to build both LoRA and textual inversion fine-tuning. I chose to use this checkpoint over stable diffusion v2 to reduce the memory requirements for training and inference.

Dataset: To begin fine-tuning the model, I manually collected a set of images of baseball pitchers throwing pitches from a variety of angles. The dataset can be found on HuggingFace here. These images were sourced from MLB's official photo stream, as well as official press releases made by teams and news organizations. To generate captions for these images for training, I used the open-source model BLIP [Li et al., 2022]. These images and their associated captions were then fed to the LoRA training method.

For textual inversion, fewer images are required. However, textual inversion requires an increased amount of preprocessing. All images fed into the model need be of a uniform size (512 x 512), and should have the subject centered and in focus. An example of the necessary data preprocessing is shown in figure 2. This preprocessing step was performed on a selection of 12 images from the dataset, and captions were generated using BLIP. However, the captions for textual inversion were edited manually, to remove any reference to the action of 'pitching', so that a chosen keyword for the textual inversion can instead be used after training is complete.

Expected results and evaluation: After training both of these fine-tuning methods, we expect to be able to generate superficially reasonable images of baseball players throwing pitches. Ideally, the generated images should be from a variety of angles, so that they are diverse enough to be used as training samples in subsequent tasks. To quantitatively evaluate the output, I used both the CLIP score (CITE HERE) [Bińkowski et al., 2021] and the Frechet Inception distance (FID). Further details of these metrics can be found in the results section.

3 Technical Approach

This section provides an overview of the mathematics and implementation behind stable diffusion, the LoRA fine-tuning method, and textual inversion. Much of this section, including the relevant diagrams, is summarized from the original LoRA paper [Hu et al., 2021a], from the diffusion paper [Ho et al., 2020], and from the textual inversion paper [Gal et al., 2022].

3.1 Stable Diffusion

At a high level, diffusion models work by first mapping input data (training images) to a continuous latent space, by iteratively adding Gaussian noise. At each layer t, the distribution is given by:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \mu_t - \sqrt{1-\beta_t}x_{t-1}, \Sigma_t = \beta_t I)$$

where the parameter β_t can be chosen in advance. Note that since we are successively adding Gaussian noise, this entire "forward diffusion process" can be done in one step, using some reparameterization tricks. As the number of timesteps $T \to \infty$, the final outcome x_T approaches a standard multidimensional Gaussian distribution. To sample from the model, the objective of the diffusion model is to learn some $p\theta(x_{t-1}|x_t)$ that reverses this Gaussian noise process. Similarly to a VAE, this is achieved by optimizing an ELBO:

$$\log p(x) \geq E_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)] - D_{KL}(q(x_T|x_0)||p(x_T))$$

$$-\sum_{t=2}^{T} E_{q(x_t|x_0)}[D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))]$$

Further refinements and alternatives to this ELBO have been proposed and shown to perform well [Luo, 2022]. Basic diffusion models use a U-net architecture that directly noises an image in its high-dimensional original space - stable diffusion refines this standard architecture by using and training an encoder to move to a lower-dimensional latent space, and using a decoder to move back to the feature space. A diagram of the architecture in latent diffusion models is shown in figure 3.

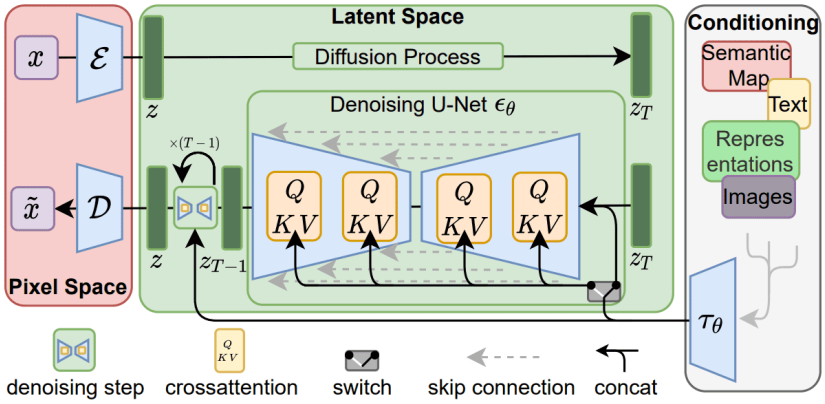

Figure 3: Stable diffusion architecture, [Rombach et al., 2022]

In Figure 3, we can observe three major components of the stable diffusion architecture. The denoising U-Net architecture is the largest trainable component of the model - it is trained on a series of noisy images (in latent space), and learns to output the 'noise' added to each image. It then generates images through a 'reverse diffusion' process, which samples from the learned noise distribution, and then successively subtracts noise to arrive at images that are close to the latent space distribution. These can then be decoded by the VAE component to return to pixel space.

The major advantage of stable diffusion is that allows for **conditional** image diffusion, where the generated output can be steered by a text prompt. Assuming we have a well-trained text embedding model (Stable Diffusion 1.5 uses CLIPText to encode text prompts into latent space), the UNet can be supplemented with crossattention layers that incorporate embedded text vectors. In Stable Diffusion, these cross-attention layers are composed of three matrices, W_q (query), W_k (key), and W_v (value). Given the encoded state of the image h_i, and the encoded text e_i, cross-attention works by calculating:

$$q_i = W_q h_i$$

$$k_j = W_k e_j$$

$$v_j = W_v e_j$$

Activations are then calculated using:

$$a_{ij} = Softmax\left(\frac{k_j^T q_i}{\sqrt{len(q)}}\right)$$

and then distributed according to the attention vector:

$$c_i = \sum_j a_{ij} v_j$$

before being reshaped and passed through the next layer of the standard U-net architecture. The model can therefore be further trained on a reconstruction loss, to tune the attention layers [Vaswani et al., 2017].

3.2 LoRA

Assuming that we already have a fully trained model checkpoint, the principle behind LoRA is injecting trainable, low-rank matrices into the model's existing weights, and only training these specific matrices. By tuning these parameters instead of the many in the complete model, we gain significant improvements in both compute time and memory. For any pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA updates the matrix using the composition:

$$W_0 + BA$$

where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$ and $r << \min(d, k)$. Only the matrices B and A are trained during LoRA, resulting in much faster optimization. In the context of cross-attention layers in neural networks, the original paper and the diffusers implementation used in this project focus on updating only the W_q, W_k, and W_v matrices in each cross-attention layer of the U-net. Training using LoRA is order of magnitude faster than training the complete model, as the number of trainable parameters is significantly reduced. In addition, training using LoRA matrices avoids the problem of 'catastrophic forgetting' [Kirkpatrick et al., 2017], since the existing representations learned by the model are never overwritten. A visualization of how the low-rank matrices fit on top of existing matrices in the model is shown in figure 4

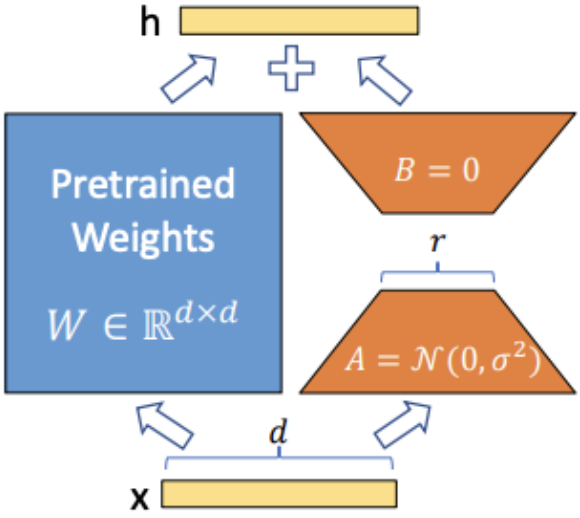

Figure 4: LoRA training, [Hu et al., 2021b]

3.3 Textual Inversion

In contrast to adding training weights to the U-net component of the stable diffusion architecture, textual inversion fine-tunes the CLIP embedding model used to convert text captions into latent space.

Using a pre-trained model, textual inversion finds new words in the model's textual embedding space [Gal et al., 2022]. By focusing on this component of the model's architecture, as opposed to retraining the entire model, similar benefits to LoRA are achieved - namely, a much smaller set of weights that can be easily distributed and added to existing model checkpoints, reduced training times, and a small set of training examples. A diagram of the process can be seen in figure 5

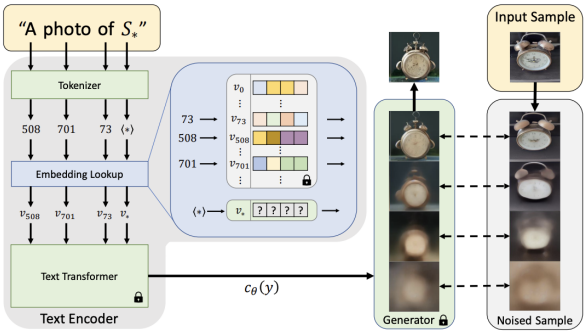

Figure 5: Textual embedding and inversion process, [Gal et al., 2022]

As seen in the diagram, a string containing a placeholder word is converted into a vector embedding $v*$. This is used to condition the generative process, and the final generative image is compared to an input sample. In order to train the model, the following optimization goal is used, to find the optimal vector embedding:

$$v* = \arg\min_{v} E_{z\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}||\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))||_2^2$$

where $\mathcal{E}$ represents the encoder, y is the conditioning input (string with pseudoword), $c_\theta(y)$ is the model that maps y to a conditioning vector, ϵ is drawn from the latent space, t is the timestep, and $\epsilon_t heta$ is the denoising U-net.

In the diffusers implementation of textual inversion, a small dataset of 3-5 images is captioned with neutral text template prompts, such as " a rendition of a $S*$" or "a photo of a small $S*$", and the result embedding look-up is optimized to minimize the reconstruction loss between the guided output and the noised sample, while keeping the models c_θ and ϵ_θ fixed.

4 Results and evaluation

4.1 Metrics

Assessing the quality of generated images is a complex task. Qualitatively, the images can be directly compared to real images in the training dataset to see how well they capture the complex and diverse pitching poses. Quantitatively, there are two primary metrics that can be used - the CLIP score for image-caption compatibility in the generated datasets, and the Frechet Inception Distance to directly compare generated images to the training images.

The CLIP score [Hessel et al., 2021] is a metric that focuses specifically on the compatibility between an image and the caption used to generate it. The CLIP model [Radford et al., 2021] is a model trained on image-caption pairs, and using CLIP with generated image-caption pairs has been shown to be highly correlated with human judgements of caption quality. The CLIP score is calculated by passing both the generated image and the caption through CLIP to obtain embedding vectors c and v respectively, and then computing their cosine similarity using:

$$w * \max(\cos(c, v), 0)$$

The CLIP score of an entire dataset is then just the average of the individual image-caption pairs. In general, a higher clip score indicates larger image-caption compatibility.

The Frechet Inception Distance [Heusel et al., 2018] uses an inception model trained on the ImageNet dataset, and compares the Gaussian distributions outputted by the activation functions in the

penultimate layer of the model. If p_w is the distribution of the real dataset, p is the distribution of the generated dataset (both assumed to be Gaussian), with means and covariances (m_w, C_w) and (m, C) respectively, then the Frechet Inception Distance is calculated by:

$$||m - m_w||_2^2 + Tr(C + C_w - 2(CC_w)^{1/2})$$

In general, a lower Frechet inception distance means that the generated images more closely resemble the distribution of the real dataset.

4.2 Results

To compare all three models qualitatively, I have generated images from each model using the same set of prompts (except for in the case of textual inversion, where the phrase "throwing a pitch" was replaced with the trained pseudo-word "<baseball-pitching>"). An example prompt and the associated images are as below:

```
Prompt:
"baseball player throwing a pitch, inside a stadium,
photorealistic, high quality, standing on a dirt mound"
```

(a) Vanilla stable diffusion (b) With LoRA fine-tuning (c) With Textual Inversion

Figure 6: Images generated with and without fine-tuning

The generated images from the base stable diffusion checkpoint (seen in figure 6a) suffer from many of common problems experienced by stable diffusion, such as missing or deformed limbs, uncentered subjects, and contorted poses. In comparison, images generated using LoRA and textual inversion (figure 6b and figure 6c) show much better subject focus, and more anatomically accurate limb positions. In general, I observed that the textual inversion model was more able to consistently produce images that resembled real pitches. However, the best generated images were typically from very similar angles as images found in the training dataset, indicating that a more diverse training set may have been needed in order for the model to fully generalize the action.

The CLIP scores averaged over a set of 6 prompts for baseline stable diffusion can be found in table 1, and the FID scores of generated images when compared to 15 sample images from the real data set can be found in table 2. It is important to note that I observed that the CLIP score is a highly sensitive metric, and that the differences between all the models are quite small. When rerunning the evaluation procedure, the ordering of the scores was not always consistent. The scores reported here are an average over 10 different generated datasets for each model. On the whole, it appears as though textual inversion has the highest image-caption compatibility, with LoRA close behind. This is in line with expected results given the quality of the generated images. However, the FID scores are more surprising - despite being superficially better, the textual inversion tuned model performed significantly worse on the FID score than even the baseline stable diffusion. There are two possible interpretations for this. The first is that the model truly did perform poorly, and the generated images did not resemble the training dataset at all, in some representation space that may not be immediately visible to an observer. The second possible interpretation is that the textual inversion model was actually able to generalize better than the other models, and therefore the distribution of generated images was significantly different than that of the true dataset, leading to a higher divergence.

Table 1: CLIP Scores

Model	CLIP score
Stable Diffusion 1-5	30.6715
LoRA tuned model	31.0790
Textual Inversion tuned model	31.3026

Table 2: FID Scores

Model	FID score
Stable Diffusion 1-5	92.65
LoRA tuned model	71.43
Textual Inversion tuned model	116.80

5 Conclusion and Future Work

Overall, it is clear that assessing the quality of generated images is a difficult process, and does not have easy interpretation. An interesting direction for future work in the context of this specific project may be using these generated images in coordination with automated human pose captioning tools, and training injury prevention/other downstream application models on a combination of synthetic and real data, in order to compare how reasonable the generated images are.

Additionally, a potential improvement of this project could be using a more complex model to caption images, as BLIP generated captions for all of the images in my training dataset were very similar to one another. This may improve the model's responsiveness to different prompts, and may help the model improve in its CLIP score. With further time and resources, I would also have been interested in investigating how ControlNet [Zhang et al., 2023] can provide more strongly guided conditional generation. ControlNet fine-tuning methods have been shown to produce highly accurate pose replication, while still being able to generate subjects in diverse backgrounds and lighting.

References

Finn Behrendt, Debayan Bhattacharya, Julia Krüger, Roland Opfer, and Alexander Schlaefer. Patched diffusion models for unsupervised anomaly detection in brain mri, 2023.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021.

Haoming Chen, Runyang Feng, Sifan Wu, Hao Xu, Fengcheng Zhou, and Zhenguang Liu. 2d human pose estimation: A survey. *Multimedia Systems*, 29(5):3115–3138, 2023.

ATLAS Collaboration. Deep generative models for fast photon shower simulation in atlas, 2022.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022.

Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, Yixiao Ge, Ying Shan, and Mike Zheng Shou. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021a. URL `https://arxiv.org/abs/2106.09685`.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021b.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

Yingzhou Lu, Minjie Shen, Huazheng Wang, Capucine van Rechem, and Wenqi Wei. Machine learning for synthetic data generation: A review, 2023.

Calvin Luo. Understanding diffusion models: A unified perspective, 2022.

AJ Piergiovanni and Michael S. Ryoo. Early detection of injuries in mlb pitchers from video, 2019.

Miroslav Purkrábek and Jiří Matas. Improving 2d human pose estimation across unseen camera views with synthetic data. 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. How robust is 3d human pose estimation to occlusion? *arXiv preprint arXiv:1808.09316*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1):1–37, 2023.